ALGORITHMIC DESIGN OF PHOTONIC STRUCTURES WITH DEEP LEARNING

A Dissertation
Presented to
The Academic Faculty

By

Zhaocheng Liu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2021

ALGORITHMIC DESIGN OF PHOTONIC STRUCTURES WITH DEEP LEARNING

Thesis committee:

Dr. Wenshan Cai, Advisor
School of Electrical and Computer Engineering, School of Material Science and Engineering
Georgia Institute of Technology

Dr. Andrew F. Peterson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Azad J. Naeemi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ali Adibi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Zhuomin Zhang
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Date approved: April 15, 2020

To my family ...

ACKNOWLEDGMENTS

I cannot begin to express my deepest thanks to my advisor Prof. Wenshan Cai, who provided me with persistent support, insightful advice, encouraging conversations throughout the duration of my graduate research, and makes everything today come true. I'm indebted to my previous and current labmates, Dr. Shoufeng Lan, Dr. Sean Rodorigues, Dr. Mohammad Taghinejad, Kyu-tae Lee, Dayu Zhu, Lakshimi Raju, and Andrew Kim. Without your guidance and encouragement, I cannot imagine myself growing so fast for the past five years. I wish the best to your future career and academic achievements. I would also like to extend my gratitude to my thesis committee Dr. Andrew F. Peterson, Dr. Azad J. Naeemi, Dr. Ali Adibi, and Dr. Zhuomin Zhang for their support.

Special thanks to my supervisors Dr. Zhaoming Zhu and Dr. Byron Taylor at Facebook. I would never have the fantastic opportunity and experience to learn the industry design and research without you. Thanks to my friends and collogues during the intern, Dr. Qing Chao, Tianyi Yang, Rishabh Gupta, Alex Göttker, Raffael Cappocia, Nathan Zhao, Jianbo Zhao, Xing Huang, and many more. Your help and support were my best memory.

I would like to extend my thanks to my advisor during my undergraduate study, Dr. Shuqi Chen, who introduced me to photonics research. I must also thank my labmates at Nankai University, Dr. Jianxiong Li, Dr. Ping Yu, Dr. Xiaoyang Duan, Dr. Zhancheng Li, Dr. Wenwei Liu, Dr. Zhi Li, and Ji Xia. I will always remember the time we spent on card games.

Thanks should also go to my friends, Yuhui Dang and Xiao Wang, who are always there helping me go through the toughest time during my graduate study. Special thanks to my friend Yuying Liu. I enjoy discussing with you about our ridiculous "theory of everything". I want to thank all my friends for their support throughout the years.

Finally, I want to thank my parents, who always encourage me to insist my own dream at any cost. I will never stop my pace to pursue the career I love in the future.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

AE autoencoder

CC cooperative coevolution

CNN convolutional neural network

CP circularly polarized

CPPN compositional pattern-producing network

DOE diffractive optical element

DOF degree of freedom

DOLP degree of linear polarization

ES evolution strategy

FCN fully connected network

FDTD finite-difference time-domain

FEM finite element method

FT Fourier transform

GAN Generative adversarial network

IFT inverse Fourier transform

IFTA iterative Fourier transform algorithm

KL Kullback-Leibler

LCP left circularly polarized

LP linear polarization

MSE mean squared error

NN neural network

RCP right circularly polarized

RCWA rigorous coupled-wave analysis

RF radio frequency

RNN recurrent neural network

VAE variational autoencoder

SUMMARY

Interactions of light with matter are the foundation of the photonic technology, which serves as a pivotal enabler for the modern prosperity we enjoy today, with gigantic amounts of information presented, processed, transmitted, and stored by optical means. The advent of photonics in the last two decades has ushered in a revolutionary means to manipulate the behavior of light on the nanoscale. The design of photonic structures, to date, has relied on the expertise of an optical scientist to guide a progression of electromagnetic simulations that iteratively solve Maxwell's equations until a locally optimized solution can be attained. In this dissertation, we identify solutions to circumvent this conventional design procedure by means of a deep learning architecture.

Our first task is to develop a generative model for the design of 2D photonic structures. Here we adopt a GAN, and constructed a geometric dataset that is comprised of images of random shapes. When training the GAN-based network with the geometric data, the produced patterns from the network resemble some samples in the geometric dataset. Given a sufficiently large geometries dataset, we are able to identify and refine the shapes of photonic structures in order to replicate the desired spectra at the input of the networks. Our design strategy allows the generation of arbitrary patterns of the unit cell structure. This approach reveals an opportunity to expedite the discovery and design of metasurfaces for tailored optical responses in a systematic, inverse-design manner.

Following the GAN-based generative design approach, we developed a framework leveraging both a deep generative model and a modified evolution strategy to automate the inverse design of engineered photonic materials and devices. Through a VAE, all potential patterns of unit structures are encoded into a continuous latent space. An evolution strategy is applied to vectors in the latent space to identify an optimized vector whose corresponding photonic structure satisfies the design objective. The capacity of the proposed methodology is tested through a case study – designing metasurfaces in response to

customer-defined spectra at the input. The evaluation shows that over 95% accuracy can be achieved for all the unit patterns of the metasurfaces in the test dataset. This scheme requires no prior knowledge of the geometry of the photonic structures, and, in principle, allows joint optimization of the dimensional parameters. As such, this work represents an efficient, on-demand, and automated approach for the inverse design of photonic structures with subwavelength features.

Molecules composed of atoms exhibit properties not inherent to their constituent atoms. Similarly, meta-molecules consisting of multiple meta-atoms possess emerging features that the meta-atoms themselves do not possess. However, the intricate mechanisms and the large degrees of freedom of the multi-element systems impede an effective strategy for the design and optimization of meta-molecules. In the next task, we proposed a hybrid framework consolidating CPPN and CC to resolve the inverse design of meta-molecules in metasurfaces. The framework breaks the design of the meta-molecules into separate designs of meta-atoms, and independently solves the smaller design tasks of the meta-atoms through deep learning and evolutionary algorithms. We utilized the proposed framework to design metallic meta-molecules for arbitrary manipulation of the polarization and wavefront of light. Moreover, the efficacy and reliability of the design strategy are confirmed through experimental validations. This framework reveals a promising candidate approach to expedite the design of large-scale metasurfaces in a labor-saving, systematic manner

At last, we proposed a topological encoding method that transforms photonic structures represented by binary images to continuous sparse representations. This sparse representation can be utilized for dimensionality reduction and dataset generation, enabling effective analysis and optimization of photonic topologies with data-driven approaches. As a proof of principle, we leveraged our encoding method for the design of two dimensional non-paraxial diffractive optical elements with various diffraction intensity distributions. We proved that our encoding method is able to assist machine-learning-based inverse design approaches for accurate and global optimization.

CHAPTER 1
INTRODUCTION AND BACKGROUND

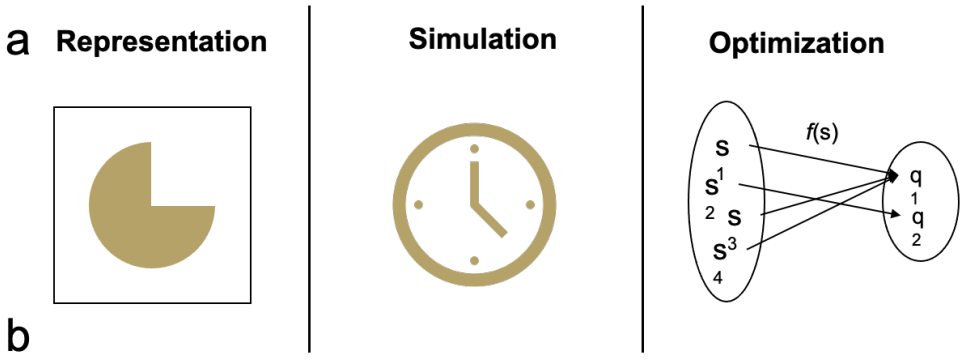
1.1 Introduction to artificially structured photonic media

Interactions of light with matter are the foundation of the photonic technology, which serves as a pivotal enabler for the modern prosperity we enjoy today, with gigantic amounts of information presented, processed, transmitted, and stored by optical means. Our ability to manipulate the light waves is largely governed by the availability of photonic materials and structures at our disposal. In particular, there is a growing trend to explore beyond naturally occurring materials or compounds and utilize artificially structured photonic media to control the amplitude, phase, polarization, and wavefront of light in unprecedented manner [1]. The rational exploitation of photonic structures with ordered arrangements of micro- or nano-scale features perhaps started with diffraction gratings, where densely spaced slits or grooves with micrometer-scale rulings were used to diffract light toward a set of wavelength-dependent directions. Over two centuries after their initial invention, diffraction gratings still serve as pivotal components in various optical instruments, such as monochromators, spectrometers, and chirped pulse amplifiers. Other grating-like, linearly periodic photonic structures include grating couplers for in/out coupling of light to optical waveguides [2, 3], fiber Bragg gratings for fiber-based communications [4, 5], sensors and lasers [6, 7], and fine metallic wires of subwavelength pitches used as polarizers thanks to the anisotropic electron confinement therein [8]. Another prominent example of engineered photonic structures is optical coatings, where multilayered thin films of appropriate thicknesses and refractive indices are employed to alter the reflection and transmission of light [9].

Over the past two or three decades, the exploration of artificially structured photonic

media, such as photonic crystals, plasmonic structures, and metamaterials, represents a central theme in the optical sciences. The emergence and continued prosperity of these novel optical media are partially driven by the combined advances in the modeling, fabrication, and characterization of complicated nanostructures, along with the unquenchable desire for an unprecedented capacity to control the flow of light at the nanoscale. Photonic crystals typically consist of repeating regions of distinct refractive indices with wavelength scale periodicity, thereby enabling allowed and forbidden spectral ranges of light, much like how electron energy bands are formed in solid crystals [10, 11, 12]. Two- and three-dimensional photonic crystals have been used in diverse applications, including optical waveguides and cavities [13, 14, 15], nonlinear signal generation [16, 17], dispersion engineering [18], etc. Structures similar to photonic crystal slabs, but made of metal, have ushered the field of plasmonics [19, 20, 21]. Plasmonics studies how light gives rise to and interacts with collective excitations of free electrons at metal surfaces, and how to best utilize engineered metallic nanostructures to manipulate light waves down to the deep subwavelength scale [22, 23, 24, 25, 26]. In addition to individually engineering plasmonic components, subwavelength metallic structures can serve as building blocks, or "meta-atoms", for optical metamaterials with properties that were not observed or even speculated about in the past [1, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36]. A metamaterial is an artificially structured material which attains its properties from the repeated unit structure rather than the constituent materials. Two-dimensional metamaterials, or metasurfaces, represent an exciting new trend in the field of metamaterials [37, 38, 39, 40, 41, 42]. By introducing spatial variations in the optical response of miniature light scatterers, metasurfaces enable arbitrary wavefront shaping with an unprecedented flexibility by producing controllable abrupt changes in the phase, amplitude and polarization of light waves at a dimension scale comparable or smaller than the wavelength of light [43, 44, 45, 46, 47, 48].

a



b

Representation	Simulation	Optimization
Parameter list	FDTD/FDFD	Bayesian optimization
Polynomial/Polygon	FEM	Evolutionary algorithm
Level sets	RCWA	Adjoint method
Pixilate image	Transfer matrix	Search algorithms

Figure 1.1: Inverse design toolkit for photonic structures. (a) Three essential parts, representation, simulation, and optimization, need to be decided based on the characteristics of the problem. (b) Lists of some common approaches for the representation, simulation, and optimization in photonic research.

1.2 Inverse design methods for photonic structures

Analogous to the subject of macroscopic artificial structures, the design of microscopic structures remains a major topic in photonic research. Although photonic structure performance is typically straightforward to predict, through sophisticated simulation algorithms such as FEM and finite-difference time-domain (FDTD), the inverse problem, designing an on-demand photonic device, is not closed-form. At the early stages of nanophotonics research, the prototypical designs were mostly based on educable guesses such as the split-ring [25, 27, 28, 29], V-shaped antenna [23, 30, 31] and gammadions [32, 33, 34] to name a few. However, limited by the prior knowledge of humans and the complicated light-matter interaction mechanisms, photonic devices with unconventional functionalities and extremely high efficiencies may have never been discovered with intuitively guessed geometries. In order to address the difficulty of photonic and optical design, algorithmic optimization methodologies, are developed for the optimization of photonic devices.

Generally speaking, the inverse design toolkit consists of three essential parts – representation, simulation, and optimization, as illustrated in Figure 1.1(a). Representation is the method of parametrizing the photonic structures so as to perform optimization. Some parametrization methods are listed in Figure 1.1(b). The parametrization methods are associated with the geometric characteristics of the photonic structures. For example, if the structure can be represented as a few parameters like widths and heights, a list containing those parameters is sufficient. However, if the topology of a photonic structure needs to be optimized, the structure should be represented by level set [49, 50] or pixilated images. In order to evaluate the performance of the devices during the optimization, fast, accurate simulation of the structures are required. There are sophisticated simulation tools, either open-source or commercial, that are able to tackle complicated modeling problems for light-matter interactions. Depending on the physical process, different simulation methods, such as FEM, FDTD, and rigorous coupled-wave analysis (RCWA), can be implemented. One last essential part of the toolkit is the optimization approaches. In photonic research, various optimization methods such as Bayesian optimization [51], adjoint methods[52, 53, 54] and evolutionary algorithms [55, 56], have become main themes in the field in recent years. These algorithms have successfully been implemented for the design of various unconventional photonic devices, such as power splitters [57, 58], light trapping structures [59, 60], and dielectric nanoantennas [61, 62]. To design a photonic structure or device, a proper combination of representation, simulation, and optimization methods should be selected. In order to further expand the capabilities of machine-aided design approaches, and to avoid some downsides of traditional optimization (such as the local minimum problem and expensive computations), the optical community has started to look at data-driven and machine learning methods as alternative approaches to address the inverse design problem.

1.3 Deep learning and its application in scientific research

1.3.1 Brief introduction to deep learning

In recent years, deep learning has become one of the most active research direction in the interdisciplinary research. As a machine learning method, deep learning has gone through decades of development and perfected itself as a stand-alone subject [63, 64, 65]. Proven to be a universal paradigm of data analysis, deep learning has found its applications in numerous disciplines, such as computer vision [66], pattern recognition [67, 68], and natural language processing [69, 70]. Modern deep learning architectures are based on neural networks, which are inspired by the learning patterns in biological nervous systems. A neural network is composed of multiple layers of artificial neurons, each performing a certain transformation on its input information. With the combined transformations throughout all layers, the neural network is essentially capable of representing arbitrary real-valued functions. Advanced optimization algorithms based on backpropagation and gradient descent also enable the neural network to adaptively change its parameters so as to effectively and progressively approximate the distribution of data. As a result, a well-trained neural network is able to extract the implicit features of raw data and reveal the fundamental laws beneath the dataset.

In the past two decades, the prevalence of information technology has enabled the accessibility of an unprecedentedly gigantic amount of data [71, 72], and the advances of GPUs have made the training of large-scale neural networks feasible [73]. These contributions from the computer science community have led to continued breakthroughs and explosive developments in deep learning since a couple of years ago. It has been demonstrated that a well-tuned neural network could outperform any other learning algorithm in image recognition [74, 75]. Afterwards, the popularization of very deep neural networks brought the performance of deep learning to new heights [76]. The well-known AlphaGo appeared as a milestone, with convincing evidence that deep learning is able to manage

complex tasks at a level well beyond that of human experts [77, 78]. Currently, deep learning serves the modern society in image recognition [79], objective detection [80, 81], self-driving vehicles [82], speech recognition [83], language translation [84], data mining [85], medical imaging [86], and much more.

1.3.2 Deep learning in scientific research

Besides commercial applications, deep learning is assuming an ever-growing importance in scientific research. It is becoming an indispensable tool for chemists to design molecules structures [87, 88], plan chemical syntheses [89], and predict the functionalities of materials [90]. Meanwhile, the biological community finds deep neural networks applicable to the detection of tumors [91, 92], the improvement of medical imaging quality [93], and the analysis of DNA sequences [94, 95]. In the frontier of physics, deep learning is capturing growing attention in facilitating scientific discoveries, with a number of reported successes in the classification of celestial bodies, the detection of high energy particles, and the investigation of many-body systems [96, 97, 98, 99, 100]. As a nascent area of research, quantum deep learning establishes a new bridge between neural networks and quantum information [101], a promising direction for the development of future artificial intelligence.

1.3.3 Fundamental deep learning architectures

Modern deep learning architectures are based on neural networks, which are inspired by the learning patterns in biological nervous systems. Generally speaking, a neural network is composed of multiple layers of artificial neurons, each performing a certain transformation on its input information. With the combined transformations throughout all layers, the neural network is essentially capable of representing arbitrary real-valued functions. In optical and photonics research, three network architectures are commonly mentioned: FCN, CNN, and RNN. The schematics of the three networks are listed in Figure 1.2.

The FCN is the primitive type of neural networks. An FCN consists of multiple layers

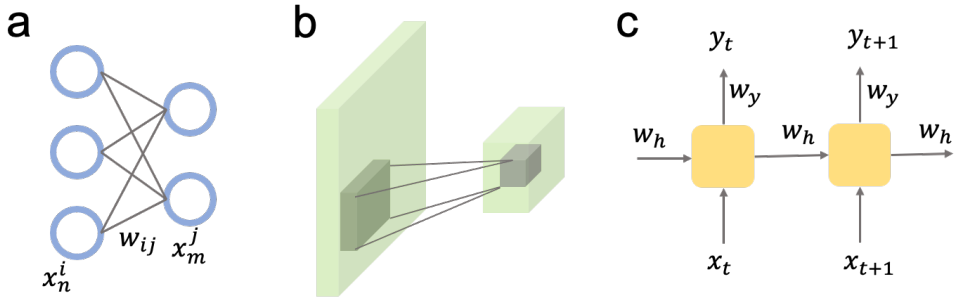


Figure 1.2: Schematics of basic structures of (a) an FCN, (b) a CNN, and (c) a RNN.

of neurons, and each neuron is connected to all the neurons in the adjacent layers. The fully connected properties provide the FCN sufficient capacity to mimic any complicated transformations. However, the dense connection also consumes large computing resources. To relax the computational cost without sacrificing the performance of neural networks, CNN is coined as an improved alternative. Instead of calculating the weights between all the connected neurons, each layer of a CNN conducts cross-correlation operations between the incoming tensor and the convolutional kernel. Such operations maintain the translation invariance over the input tensor, enhancing the efficacy of capturing features from image/audio data with strong spatial/temporal correlations. When it comes to sequential data, RNN are the paradigm that is mostly used. RNN can be explicitly generalized as sequentially connected neurons. An RNN is able to process the input data at different timestamps one at a time and generate a sequence of data based on the input time series.

1.4 Overview of deep learning-enabled inverse design for photonic structures

Recently, the optical community has been progressively migrating the techniques of deep learning and data science into photonics research, with a number of successful applications including ultrafast optics [102], optical communication [103, 104, 105], and optical microscopy [106]. On the other hand, researchers are seeking methods from photonics to solve machine learning problems. An exciting interdisciplinary example is the development of optical analog modules to accelerate the mathematical operations in deep neural

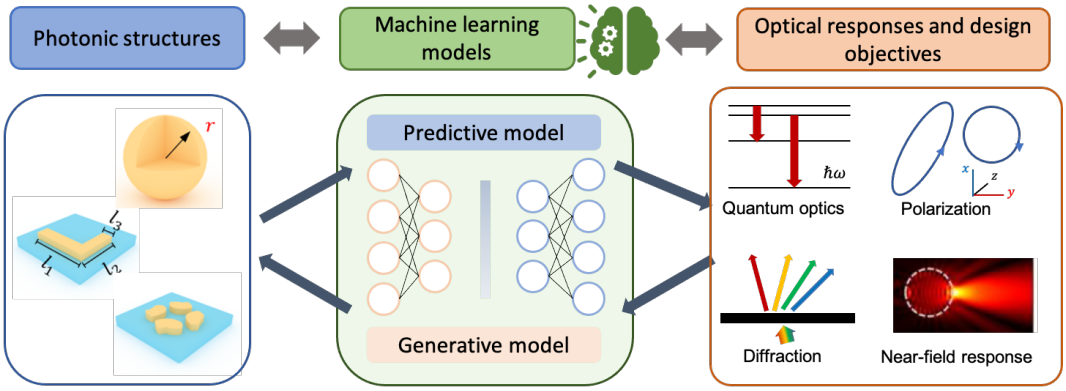


Figure 1.3: Incorporating machine learning into the design of photonic structures. Given a photonic system, a machine learning model is able to predict its optical responses, and help to identify the optical parameters defining the photonic structure.

networks by leveraging nanophotonic circuits [107, 108] and diffractive optics [109, 110, 111]. With the astronomical capability of capturing essential features from vast amounts of high dimensional data, machine learning models have become a promising tool to aid photonic design in various ways. As illustrated in Figure 1.3, given an optical system or photonic structure, deep learning models can be developed to perform the forward modelling, i.e. predicting optical responses of the given system. With proper design objectives and optimization approaches, deep learning can also help the inverse design of the optical systems. In this section, we will discuss the general design strategy using deep learning and introduce the recent progress on the deep-learning-assisted photonic inverse design.

1.4.1 General design strategy using deep learning

The machine learning methodologies used in photonic inverse design are associated with the DOF of the photonic structures. We have listed the general implementation of machine learning models in various design tasks in terms of the DOF as shown in Figure 1.4. Traditionally, with only a few parameters to optimize as in Figure 1.4(a), analytical calculation and parametric sweeping are sufficient. With an increased DOF, the optimization space grows fast and simple parametric sweeping cannot yield a satisfactory result. In order to accelerate the optimization process and explore new designs, discriminative models have

been proposed to assist the design as shown in Figure 1.4(b). Typically, discriminative models are used to learn the bidirectional mapping of the optical structures and their optical responses. However, since multiple distinct optical structures may correspond to an identical optical response (which is also called the degeneracy problem), the mapping from optical responses back to the structural parameters requires additional processing. Several proposed methods that can avoid the degeneracy problem will be listed in a later section.

When the DOF continues growing to thousands and more, the immense dimensionality of the optimization space invalidates the approaches that require huge amounts of data or vast iterations of simulations. Generative models, in this case, can be leveraged to reduce the dimensionality of the design structures and optical responses. As shown in Figure 1.4(c), a trained generative model is able to encode the photonic structure data into a compact representation, and to produce unlimited photonic structures by feeding reduced dimensional vectors to the model. Optimal design can be exhaustively searched from the compact representation. Besides, because the generative model is essentially transforming the original data to another representation, the local minima in the original parameter space are also varied and possibly eliminated in the transformed one. This property may alleviate the local minimum problem if optimization is performed in the sparse representation. Another usage of the generative model is to encode high dimensional data into a compact representation, to provide insights on the relations among data to assist the optimization. It has also been proven that generative models can be incorporated to the traditional optimization algorithms to assist fast design and avoid local minimum problem.

1.4.2 Recent progress on deep-learning-assisted photonic inverse design

Very recently, researchers have successfully implemented deep learning techniques in the inverse problem in photonics and optics. For example, metasurfaces and metamaterials that manipulate the transmittances, polarization, and circular dichroism can be identified through a bidirectional neural network [112, 113, 114, 115]; layered structures, including

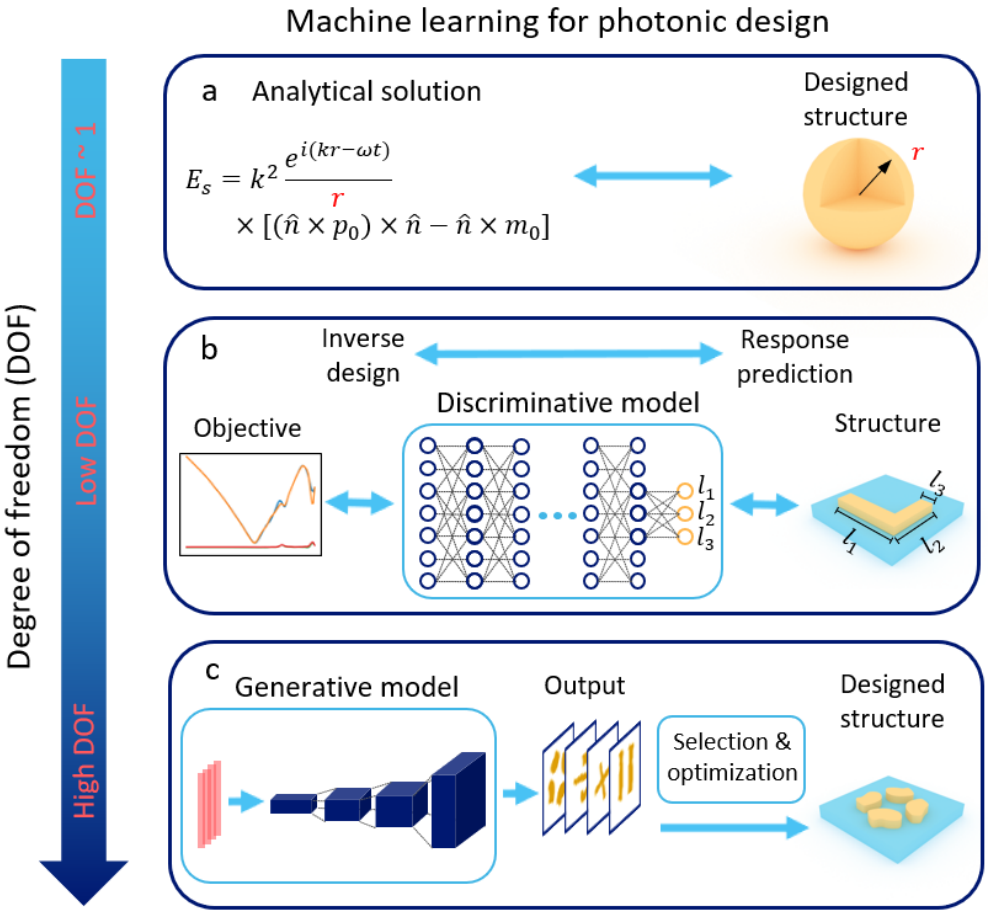


Figure 1.4: Methodologies of photonic design through machine learning at different DOF. (a) When the DOF of the photonic structure is low, the optimal combination of the design parameters can be found by analytical solution or simple parametric sweeping. However, such a design strategy may not yield optimal performance. (b) As the DOF of the design grows, the solution space expands. Without a proper optimization algorithm, the identification of optimal designs requires exponentially increasing iterations of simulation when the dimensionality of the design space grows. In the context of machine learning, we may use discriminative model to capture the relations between design parameters and optical responses with substantially reduced amount of data. It should be noted that, since multiple configurations of structures may correspond to the same response, a single discriminative model is not able to perfectly map an optical response back to a unique set of design parameters. Additional training strategies are required if discriminative models are used for the optimization and design. (c) When the DOF continues growing to thousands and more, generative models can help to reduce the dimensionality of the design and to seek relations between design parameters and optical responses for further optimization. The generative models can be jointly leveraged with discriminative models as well as traditional optimization algorithms to speed up the design process or to locate the global optimal solutions.

thin-film structures and spherical nanoparticles, can be optimized through deep learning approaches with an expedited and parallelized manner [116, 117, 118, 119, 120]. Such photonic structures have promising applications in color generation, scattering manipulation, and analog computing. In order to design high DOF photonic structures such as integrated photonic power splitters [121] and photonic crystals [122] , CNN are leveraged for the modelling and design of the systems. Advanced network architectures such as GAN are implemented for the discovery of highly complex photonic structures such as radio frequency (RF) metasurfaces [123, 124], power splitters [125], integrated photonic devices [126], and optical invisible cloaking [127]. It has also been reported that deep learning and traditional optimization approaches can be combined to globally identify optimal photonic structures, such as meta-gratings and thermal emitters [128, 129, 130, 131, 132]. In addition to the aforementioned work, there are a number of excellent research themes that we have not covered. Some complete summaries of the recent progress have been provided in a few review articles [133, 134].

1.5 Dissertation overview

The dissertation will be arranged in the following manners. In Chapter 2, a deep learning method for modelling photonic structures is introduced. Several techniques that mitigate the reliability of large dataset and improve the convergence speed will be discussed. In Chapter 3, a GAN-based framework for the inverse design of 2D nanostructures in metasurfaces is introduced [135]. The efficacy of such a strategy is demonstrated through a few meta-atom designs that control light transmittances. Chapter 4 describes a framework that incorporates evolutionary algorithms into the generative model to expedite the optimization process and improve the design fidelity [136]. In Chapter 5, the developed framework is augmented with sophisticated network architectures and optimization algorithms. Several design examples that manipulate light polarization and propagation will be presented [137, 138]. Chapter 6 describes an encoding method for the data-driven design of photonic

structures [139]. It has been implemented for the design of DOEs with stringent diffraction angles and intensities. The final Chapter provides a summary of the dissertation. Challenges of deep-learning-enabled photonics design and future research directions will also be discussed.

CHAPTER 2
CONVOLUTIONAL NEURAL NETWORK FOR EFFICIENT SIMULATION OF PHOTONIC STRUCTURES

As an essential part of the inverse design toolkit, simulation methods determine the speed of the optimization and fidelity of the design. When simulation data can be retrieved in a relative efficient manner, training a surrogate model that fast approximates the physical responses of a device is always beneficial. Deep learning model, with the capability of approximating complicated functions, is a great candidate as a surrogate model for the photonic simulation [140, 141]. The accuracy of the surrogate model directly affect the fidelity of the design and thus determines the additional efforts of post-processing the design. In this chapter, a deep learning-based modeling method and its improvement will be presented. The content of this chapter is assembled from multiple published works of the author from the past two years.

2.1 Predicting optical responses of photonic structures with CNN

When modeling photonic structures with complex topologies, we usually represent the structures in binary images. However, using a traditional statistical surrogate model to approximate physical properties with an input of image is problematic. This is because the dimensionality of the input is huge, while no regression models can accurately predict the responses given their limited capacity. Thanks to the development of modern deep learning techniques, CNN can be utilized as a surrogate model to replace the rigorous simulation of complex photonic systems. Each layer of a CNN conducts cross-correlation operations between the incoming tensor and the convolutional kernel. Such operations maintain the translate invariance over the input tensor, enhancing the efficacy of extracting features from image data with strong spatial correlations. It has been proven that CNN has

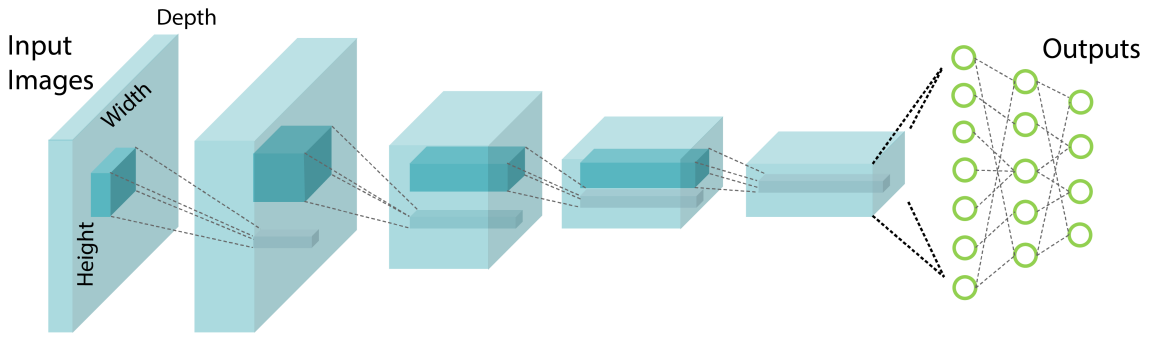


Figure 2.1: Illustration of a CNN simulator. The simulator is constructed from a sequence of convolutional layers and three fully connected layers. The simulator takes a binary image representing the topology of photonic structures as the input, and approximates the spectral components T_{xx}, T_{yy}, T_{xy} and T_{yx} .

an outstanding ability to capture the complex light-matter interactions, and approximate the optical responses of complex photonic structures with sufficient accuracy. In the context of inverse design of photonic structures represented in binary images, a CNN simulator can be trained to expedite the simulation process, and thus lead to a faster convergence to optimization results.

As an example, we trained a CNN simulator to efficiently model metasurfaces for various optical responses. The gold-patterned meta-structures in our examples are represented in 64 by 64 binary images and the four-component complex transmittances (or reflectances) T_{xx}, T_{yy}, T_{xy} and T_{yx} are stored in vectors of dimension 32. The periodicity of the lattice is $p = 320\,\text{nm}$ and the thickness of the pattern is $t = 40\,\text{nm}$. The basic architecture of a CNN simulator is illustrated in Figure 2.1. Three fully connected layers are cascaded after a CNN to approximate spectral responses T_{xx}, T_{yy}, T_{xy} and T_{yx}, each of which is represented in a vector with a dimension of 32. We create a dataset with 10,000 FEM-simulated photonic structures, 8,000 of the which are used for training and the rest for validation and testing

After training the simulator 100 epochs with a learning rate of 1×10^{-5}, the simulator is able to achieve an accuracy of 95% (defined as mean squared error (MSE)) on the validation and test dataset. Figure 2.2 presents four examples of the approximated spectra given unit

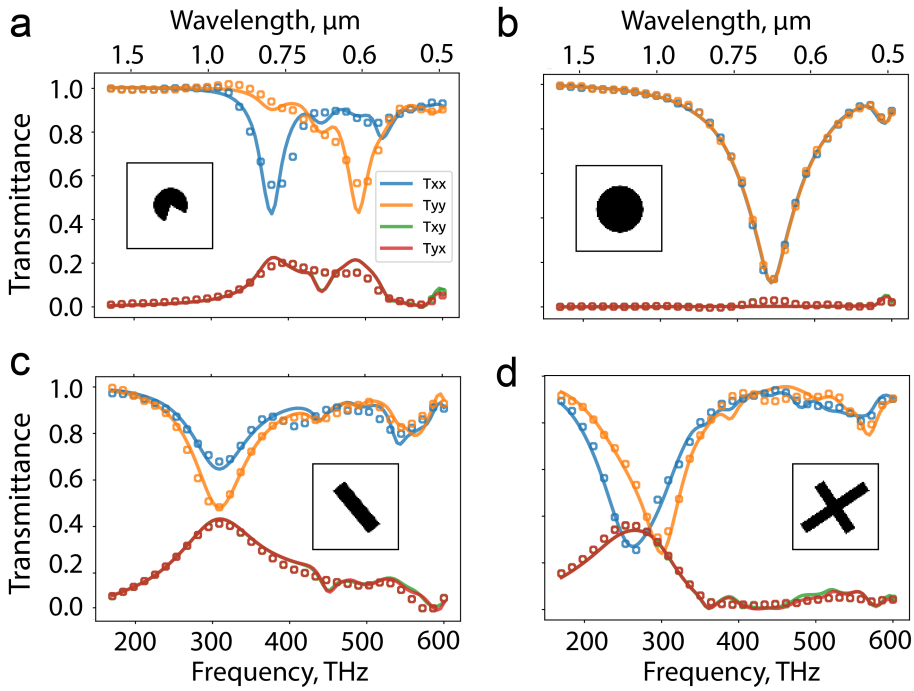


Figure 2.2: Performance of the network simulator. (a) – (d) Four examples of spectra T_{xx}, T_{yy}, T_{xy} and T_{yx} approximated by the network simulator. The input of the network are unit cells of metasurfaces as shown in the binary pattern in each plot. The spectra predicted from the network and simulated by FEM are represented by circles and solid lines, respectively.

cell of metasurfaces randomly selected from the test dataset. The predicted spectra are in agreement with the FEM-simulated ones with minimized discrepancy. As we will show in a later chapter, given a generative model is also utilized, such a prediction accuracy is sufficient for the inverse design of multi-functional metasurfaces.

In addition to training the simulator with gold-patterned metasurfaces, we also train a simulator for silver metasurfaces using the transfer learning technique. Particularly, we prepare a smaller dataset with a total number of 2,000 simulation data points for silver metasurfaces. The settings for the silver metasurface simulation keep the same as the ones for gold metasurfaces. We replace the last few fully connected layers of the trained simulator for gold metasurface with randomly initialized layers and train them using the new silver metasurface dataset. After 100 epochs of training, the accuracy of the overall network achieves 96%. The success of training of the network with small dataset relies on the geometric features extracted by the CNN layers. These features are sufficient for the new

task with similar objectives. This indicates that a CNN simulation can be readily translated to other scenarios without repeated preparation of extensive dataset.

2.2 Improving the modeling accuracy

Although it has been proven that deep learning models are able to capture the essential features of optical systems, and yield sufficiently accurate predictions of optical responses, we still need to produce large datasets so as to fit the thousands and millions of parameters in the deep learning models. This is not always possible in practice, given the expensive electromagnetic simulation and the limitation of computing resources. To enhance the accuracy of the approximation and reduce the reliability of massive dataset, various techniques have been explored. Here three techniques will be discussed, including augmenting the dataset, adopting advanced network architectures, and enforcing the boundary condition in the deep learning models.

2.2.1 Data augmentation

Although it seems that deep learning models are able to learn the relation between the photonic structures and their optical responses, the reason that utilizing a deep learning model as a surrogate model in photonic research is not always convincing. In photonic research, training data are usually generated from simulation, and the size of such generated data is limited, usually in the order of 10^4 to 10^5. Compared to the number of parameters in the modern network architectures, which are in the order of millions, the data size is significantly smaller, and consequently, the machine learning models are very likely to be overfitting. Thus, some data augmentation methods should be used.

When dealing with the image data that represent photonic patterns, we can randomly shift, flip, and rotate the patterns in the image. Shifting the patterns does not affect the optical responses, while flipping and rotating the image may change the actual physical properties. The optical responses of the flipped structures can be directly inferred from

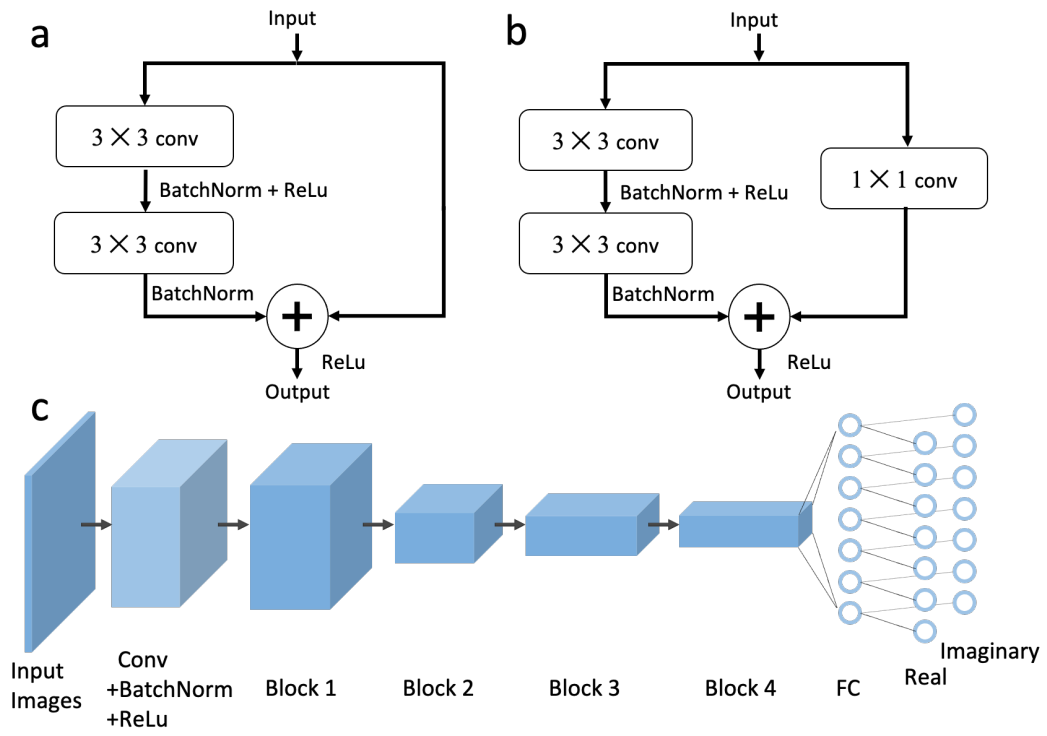


Figure 2.3: Illustration of a ResNet simulator network. (a) and (b) Two types of Resnet blocks implemented in the simulator. (c) An illustration of the architecture of the simulator network. The simulator takes the patterns of photonic structures and predicts the real and imaginary parts of their farfield spectra.

the original images through the symmetry of the physical systems. When it comes to rotation, if the photonic structure can be formulated as a linear transformation of the input, the output of a rotated structure is likely to be directly derived. For photonic structures that are represented by binary images, we can also randomly pick different two values to represent the regions with and without structures. With random shifting, flipping, rotation, and picking binary values threshold, we essentially create infinite number of input/output training pair, and thus greatly reduce the overfitting problem. It is noteworthy that we can also perform affine and perspective transformations to the input. The optical responses cannot be easily inferred in this circumstance, but the two transformations can be used as the augmentation methods when we are training generative models to produce infinite candidates of the photonic structures for optimization. This topic will be covered in a later chapter.

2.2.2 Network architecture

In addition to standard CNN networks, we can use advanced network architectures, such as residual network (ResNet) [79], to improve the accuracy of prediction. ResNet is a network structure that allows skip connections between two layers. This skip connection solves the problem of diminished gradient during training in ultra-deep networks. Figure 2.3 (a) and (b) illustrate two different blocks used in the ResNet. The input tensor is sent to two sets of CNN transforms, and the transformed tensors are combined at the output. The detailed structure of ResNet block can be decided based on the specific problem. We built ResNet simulator as shown in Figure 2.3 to serve as the surrogate model. The input photonic structures are processed by a sequence of ResNet blocks, and an FCN is concatenated to predict the complex transmittance of the spectra based on the features extracted from the ResNet. Empirically, we can observe expedited convergence rate and slightly improved accuracy with advanced network architectures. The model robustness against the input noise is also significantly improved. This indicate that, if the data set is sufficiently

large, we may always adopt the state-of-the-art deep learning models from computer vision research. However, this method does not take into account any physical information, neither does it provide any physical insights. In applied science and engineering, we may consider advanced networks for an improved prediction accuracy, but in rigorous physical problems, we should always stick to fundamental network architectures.

2.2.3 Boundary condition

Periodic boundary conditions are one of the most commonly seen boundary conditions in large-scale photonic systems. In such a system, certain photonic structures, or building blocks, constitute the overall system by spatially repeating themselves. Some examples include photonic crystals, metasurfaces, and topological photonics. An example of a photonic systems with periodic boundary is shown in Fig . 2.4(a). The system is composed of three different types of particles. Each building block, indicated by the red square, contains each of the three particles with a fixed relative spatial position. However, the choice of a unit cell is not unique. We may shift the red square to the blue, green, and black squares, and it is still a valid unit cell. As indicated in Fig 2.4(b), the four choices of unit cell are physically identical, although visually distinct. One common feature of all the cells is that the left boundary is connected to the right one, and the upper boundary is identical to the lower one. This is the periodic boundary condition.

When using CNN to process the images that contains periodic photonic structures, the periodic boundary condition is not considered. Physically, this indicates we do not take into account the interaction between two adjacent cells. We can alleviate this problem by randomly shifting the training data across the boundary, but it does not solve the fundamental restrictions of CNN applied to periodic structures. Therefore, we have to revise the operation of CNN to enforce the periodic boundary condition.

The basic operation of a CNN layer is to perform autocorrelation between a kernel and input image. In some cases, additional pixels are concatenated to the image to revise the

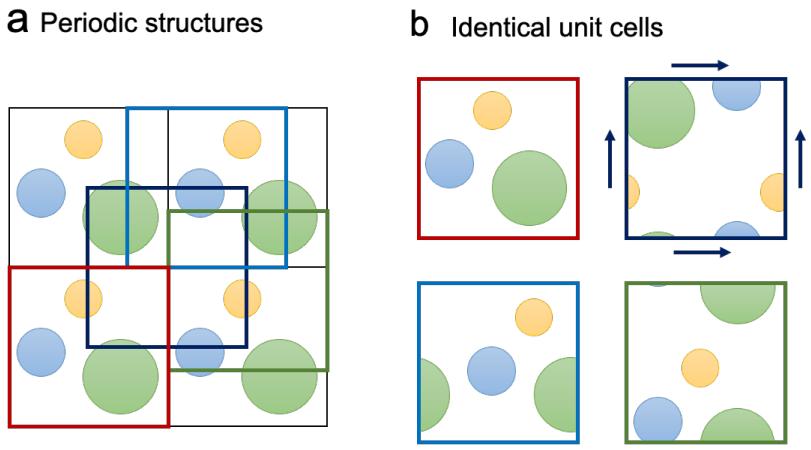


Figure 2.4: Illustration of a photonic structure with periodic boundary condition. The unit cells in (b) corresponds to the the ones in (a) with the same boundary colors. The patterns in the different unit cells are different, but they are identical in terms of optical responses.

input size. This process is called padding. According to the properties of autocorrelation, if the output of a CNN is required to be a periodic image, we can add a cyclic padding to the input image, perform the autocorrelation transform, and crop the center part of the transformed image. The cropped image is naturally a periodic image. In a CNN simulator, we can replace all the CNN layers to the padding-transform-cropping operations to enforce the boundary condition. Because the revision of the CNN layer does not incur computing burden, the overall training and inference speed are not significantly affected.

With this slight change in the CNN layer, the interaction between adjacent cells are considered. The CNN simulator is then able to fast capture the underlying relations between photonic structures and their optical response. This dramatically reduces the required size of dataset and speed up the convergence rate during training. We observed a faster convergence with smaller dataset in all our experiments. However, the improvement of the accuracy is not observed on a large deep learning model trained with a massive dataset. This is because a large deep learning model, when data augmentation is carried out, is able to recognize all the training cases. The revised CNN simply expedite this process. Since it is not reasonable to benchmark the improvement given distinct physical problems, we present a few generated photonic patterns using the revised CNN shown in Fig 2.5. On the

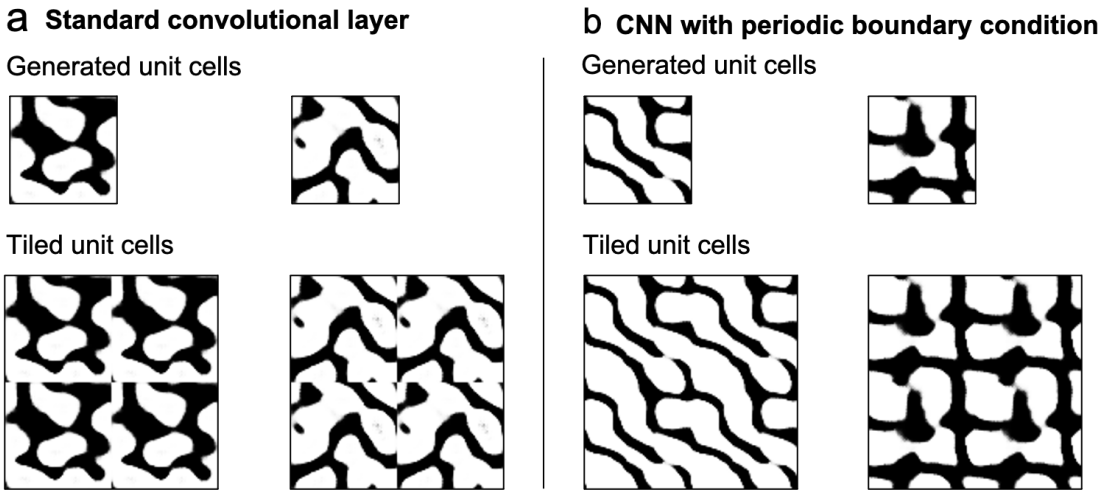


Figure 2.5: Generated patterns from a standard CNN and a CNN with periodic boundary condition. (a) Generated patterns from a CNN. The tiled patterns show discontinuities on the boundaries of the cell. (b) Generated patterns with implementation of boundary conditions in the CNN. The topologies of the patterns show periodic connectivity. This indicate that the revised CNN takes into account the interaction between adjacent unit cells in the modelling of photonic structures. It is noteworthy that in both cases the training dataset are patterns with periodic boundary condition.

left, the pattern is generated from standard CNN, while on the right the unit cell is derived from the revised CNN. We can see that the tiled unit cell of the left one does not maintain periodic boundary condition, while the right one shows continuous connection between adjacent cells. It should be noted that the training images are all periodic patterns.

CHAPTER 3
GENERATIVE MODEL FOR THE INVERSE DESIGN OF METASURFACE STRUCTURES

3.1 Introduction to generative models

3.1.1 Discriminative and generative models

Machine learning models can be roughly categorized into two classes, discriminative models and generative models. Mathematically, a discriminative model predicts the probability of label y conditioned on the input data x, i.e., $p(y|x)$, while the generative learns a probability distribution of x, $p(x)$ or a joint probability of x and y, $p(x, y)$. Intuitively, a discriminative model is a function $f(x)$, transforming the input data into a label or value $y = f(x)$. Thus, as shown in Figure 3.1, discriminative models are always related to classification and regression tasks in supervised learning schemes. A generative model, on the other hand, can map the data x into a compact representation, and with some sample algorithm, we can retrieve more data that is similar to the input dataset. Generative models usually serve in the unsupervised learning paradigms, such as classification without labels. It should be noted that the introduction to generative and discriminative models is not rigorous or complete, and there are different interpretations of the two models.

Regarding the implementation of the two models, if we use deep neural networks, we can say the models are deep discriminative/generative models. In specific, deep discriminative models are typically a direct implementation of fundamental network architectures such as an FCN or CNN. With sufficient training data, which could be parameters of the photonic structures, and labels, which could be spectral responses, we can have the modern deep learning frameworks take care of the training of the network. However, the implementation of generative models is not as straightforward as discriminative models, as gen-

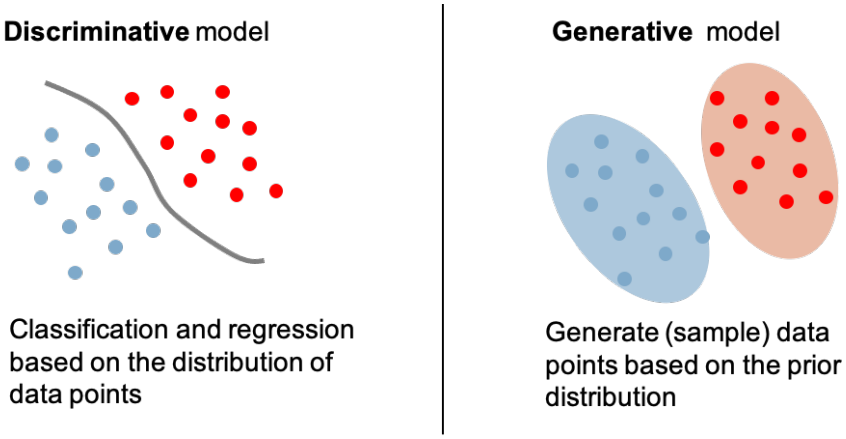


Figure 3.1: Illustration of discriminative models (left) and generative models (right).

erative models are built upon several interactive network modules. The essential usage of deep generative models in photonic research is to either capture the distribution of datasets to provide insights of design, or to perform dimensionality reduction to simplify the optimization. In a latter section, we will have a very brief introduction of generative models that have been used in photonics research with concrete examples.

3.1.2 VAE and GAN

Identifying the topology of a photonic structure given certain optical responses is a one of the central research topics in the inverse problem in optics. The topology is usually represented by binary images, the parameters that define an image can easily grow up to thousands and even millions. The high dimensionality of the photonic structure invalidates most of the optimization approaches to identify the global optimal solutions. One of the solutions that alleviates the problem is utilizing deep generative models to encode the photonic structures into a low-dimensional space. Optimization can be performed on the compact encoded space, and optimal solutions are more likely to be identified after redundant information is filtered out.

Two of the most fundamental and widely used deep generative models are VAE [142, 143] and GAN [144]. Figure 3.2 shows the schematics of the two network architectures. A VAE consists of an encoder and a decoder. The encoder transforms the input into a

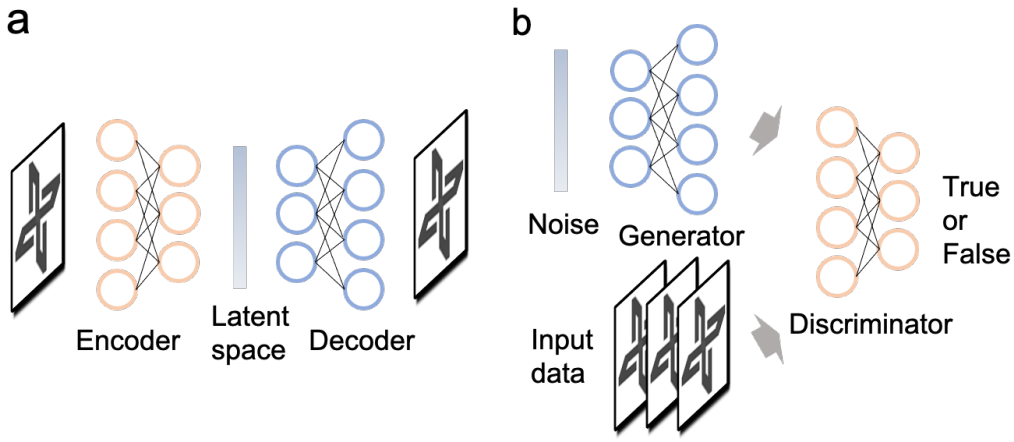


Figure 3.2: Schematics of a VAE (left) and a GAN (right).

low-dimensional space. The distribution of the encoded data is predefined as Gaussian distribution, and is parametrized by two vectors, representing the mean μ and the standard deviation σ of the distribution. The decoder can sample a vector from the distribution, and transform the vector to an original image. A GAN is composed of a generator and a discriminator. The generator transforms a noise vector into a photonic structure. The discriminator accepts both the training data and the generated data, compares the difference of the two sets of data, and guides the generator to produce images that are similar to the training data. In some cases, the discriminator is also called a critic because it provides information to correct the behavior of the generator. The detailed training method will be discussed in the later chapters where they are introduced to the optimization framework.

Although the architectures, performance, and training approaches of GAN and VAE differ, both GAN and VAE are able to capture the distribution of high dimensional data and represent them in a reduced dimensional space, which is also called a latent space. By sampling the latent space, generative models can produce more data that are "similar" to the training dataset. Since the two models are equivalent in terms of encoding photonic structures, we will always name the network that transforms a latent vector into a photonic structure as the "generator".

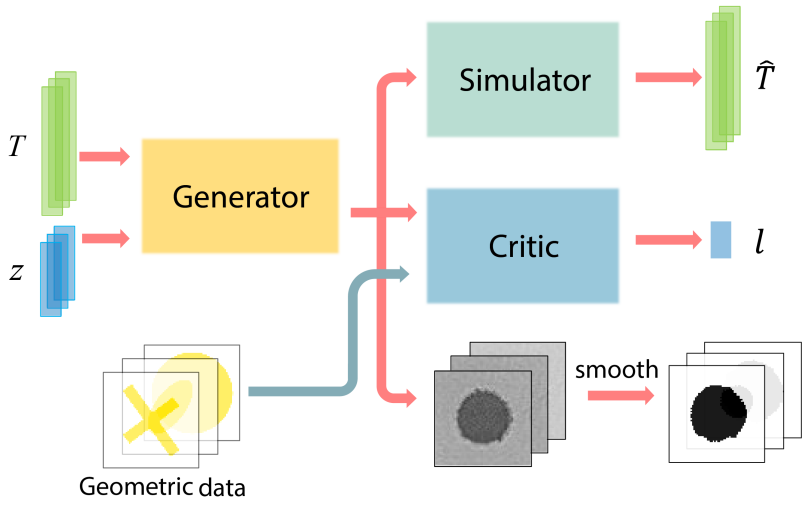


Figure 3.3: Architecture of the proposed network for optical design. Three networks, the generator (G), the simulator (S), and the critic (D) constitute the complete architecture. The generator accepts the spectra T and noise z and produces possible patterns. The simulator is a pretrained network that approximates the transmittance spectrum for a given pattern at its input, and the critic evaluates the distance of the distributions between the geometric data and the patterns from the generator. While training the generator, the produced patterns vary according to the feedback obtained from S and D. Valid patterns are documented during the training process, and are smoothed to qualify as candidate structures.

3.2 GAN-based design strategy for metasurfaces

Our first task is to develop a deep learning strategy to design metasurface nanostructures that achieve desired optical transmittance and reluctance. We have constructed a GAN-based network architecture as illustrated Figure 3.3 to achieve the inverse design problem. We divide the network into three parts: a simulator (S), a generator (G) and a critic (D). The primary goal is to train an overfitted generator, which produces metasurface patterns in response to given input spectra T such that the Euclidean distance between the spectra of the generated pattern T' and the input spectra T is minimized. All three networks are convolutional neural networks with delicate differences in detailed structures. The simulator is a pretrained model with fixed weights, taking the generated patterns as input and approximating their transmittances spectra without the use of electromagnetic simulations.

It is built to control the accuracy of the optical spectra of the generated structures when training the generator. The generator and the critic together constitute a GAN. The critic of this GAN accepts both the user-defined geometric data and the patterns generated from the generator, and then yields a value l which is essential to compute the distance between the distributions of the two sets of data. By minimizing this distance, the critic network guides the generator to produce patterns that share commons features with the input geometric data. During the training process, we update the weights in the generator by backpropagation from the losses defined by the simulator and the critic. Valid patterns produced by the generator are documented throughout the training process; this occurs whenever the losses of the simulator and critic are sufficiently small. The generated patterns are finally smoothed to binary images as candidates of the metasurface design.

In essence, the critic learns the distribution of the geometric data and restricts the generator to produce patterns in the image space where the geometric data resides in. Although it is theoretically possible to define a function to replace the critic network, it is impractical to find an explicit expression for the function when the geometric data are diverse and complex as in our case. There are several benefits of including such a network critic into the overall network architecture. First, it excludes a large number of unacceptable patterns that are deemed unrealistic in actual nanofabrication. Second, it allows us to control the overall shape of the generated patterns by feeding similar geometries into the critic. While such forethoughts are not necessary for AI-guided optical design, as demonstrated in the later part of this work, certain geometrical constraints at the input may help to narrow down the potential candidates and thereby expedite the convergence to a solution. Finally, since the production of various categories of geometric data is straightforward and largely effortless, we can modify the distributions of the geometric data to avoid degeneracy. This may occur when multiple metasurface patterns of different topologies possess identical optical spectra to the input spectra, within an acceptable margin of deviation.

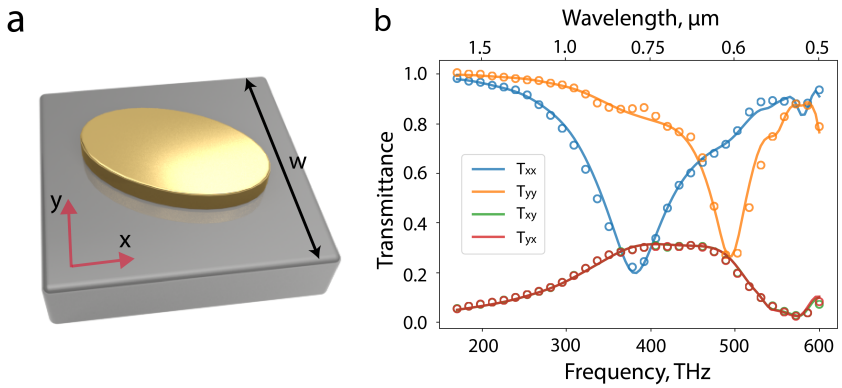


Figure 3.4: (a) The unit cell of the metasurface used in our case study of the inverse design problem. The pattern is allowed to vary for a desired spectral response, with the following structural and material constraints enforced: gold for the pattern and glass as the substrate; unit cell size $w = 340\,\mathrm{nm}$; thickness of gold $d = 50\,\mathrm{nm}$. (b) Transmission spectra of a representative structure shown in (a), obtained by FEM electromagnetic simulation (solid lines) and by the simulator network (circles), respectively.

3.3 Metasurface configuration and data representation

As a representative and generalizable case study, we applied the strategy outlined above to the design of metasurfaces with prescribed spectral behavior under linearly polarized illumination. The general unit cell of the metasurface used in this model is shown in Figure 3.4(a), which has a single layered gold pattern in a square lattice, situated on a glass substrate. Other common parameters include a lattice constant of $w = 340\,\mathrm{nm}$ and a thickness of the gold layer set to $d = 50\,\mathrm{nm}$. To train the simulator with sufficient data, we carried out 6,500 full wave FEM simulations for metasurfaces with a wide variety of shapes that replaced the metal in the unit cells. The simulation was run over a frequency span from $f = 170\,\mathrm{THz}$ to $600\,\mathrm{THz}$ (i.e., $500\,\mathrm{nm}$ to $1.8\,\mu\mathrm{m}$ in the wavelength domain), which covers a major portion of the visible and the near-infrared spectral range. These FEM simulations yield the transmittance magnitude spectra of each metasurface under x- and y-polarized illumination, where i and j indicates the polarization directions for the incidence and the detection, respectively. Throughout the training, the unit cell structure is represented as a binary image of 64×64 pixels, in which 1 stands for gold and -1 for void

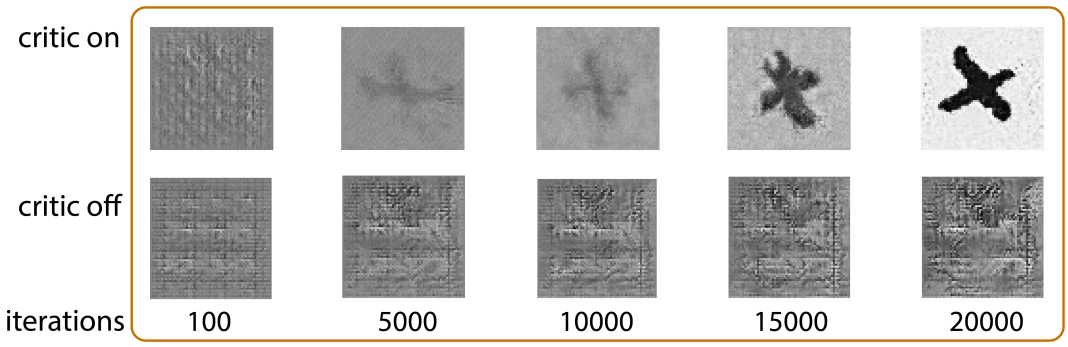


Figure 3.5: Generated patterns during the training process after certain iterations, with and without the critic network. Only geometric data categorized as a cross are fed into the critic in this example.

(air). Each transmission spectrum is represented as a 32-entry vector with equal frequency intervals. The simulator after the training process is able to approximate the transmittance T with an average absolute error of less than 0.01 at each frequency point. An example for the effectiveness of the simulator is illustrated in Figure 3.4(b), where the solid lines are the results from the FEM electromagnetic simulations for the ellipse particle array shown in Figure 3.4(a). Here, the circles represent the spectral approximation of the same structure obtained by the neural network.

3.4 Validation of the design strategy

3.4.1 Effect of the critic

To demonstrate the mechanism and performance of the network, we classify the geometric data into several classes such as circles, arcs, crosses, ellipses, rectangles, etc. Figure 3.5 illustrates a series of generated patterns during a training process after certain iterations, with and without the critic network. For this example, the critic is fed with a class of geometric data only comprised of crosses. When the critic is on, a cross pattern emerges and gradually adjusts itself to meet the requirement of the input spectrum. In sharp contrast, when the critic network is turned off, the generated pattern is a cluster of random pixels and stays stabilized after a few hundred iterations.

3.4.2 Validation with existing metasurfaces

Our network architecture is capable of generating metasurface patterns in response to an arbitrarily-input, on-demand set of optical spectra, whose components and frequency range of interest are defined by the user. At the input of the generator, we specify a set of transmittance spectra from $170\,\mathrm{THz}$ to $600\,\mathrm{THz}$. In the following discussion, we define a set of patterns to be used as a test set and denote this test set as s, the spectra of each s as T. Once these spectra are passed through the network architecture, a pattern is retrieved which we denote as s' in correspondence to s. In verification, the generated set s' are FEM-simulated and defined as T'. Unless mentioned otherwise, in the following experiments we set the number of patterns being parallelly searched at each run to be 40 and the total iterations of the training to be $50,000$. For each target spectrum, valid patterns may occur at different stages of the training process whenever the losses of the simulator and critic are both reasonably small. On a machine with a single GPU Quadro P5000, it takes approximately 10 minutes to carry out 10,000 iterations of training on average.

As an initial demonstration, to illustrate the overall competence of the proposed framework, we use the spectra, T, of randomly selected test samples from each geometric class as the input, and allow the network to seek proper patterns based on these spectra. In this situation, we ensure the existence of solutions by using actual spectra of real patterns as the "target" or input. This constraint will be removed when we perform the inverse design for on-demand spectra, as presented in a later part of the paper. In each test, the critic is fed with 1000 data points (i.e., geometrical shapes) that are randomly generated from the same geometric class. Figure 3.6(a) shows representative samples from such experiments for each class of the geometry. The first row depicts the test samples s, and the second row shows the corresponding samples s' generated by the network. The geometric patterns in each pair of the two rows agree very well, partially because the full spectra input to the generator substantially narrows down the possible solutions. Since the generator does not receive any direct information of the geometry for the input spectra, it may uncover equiv-

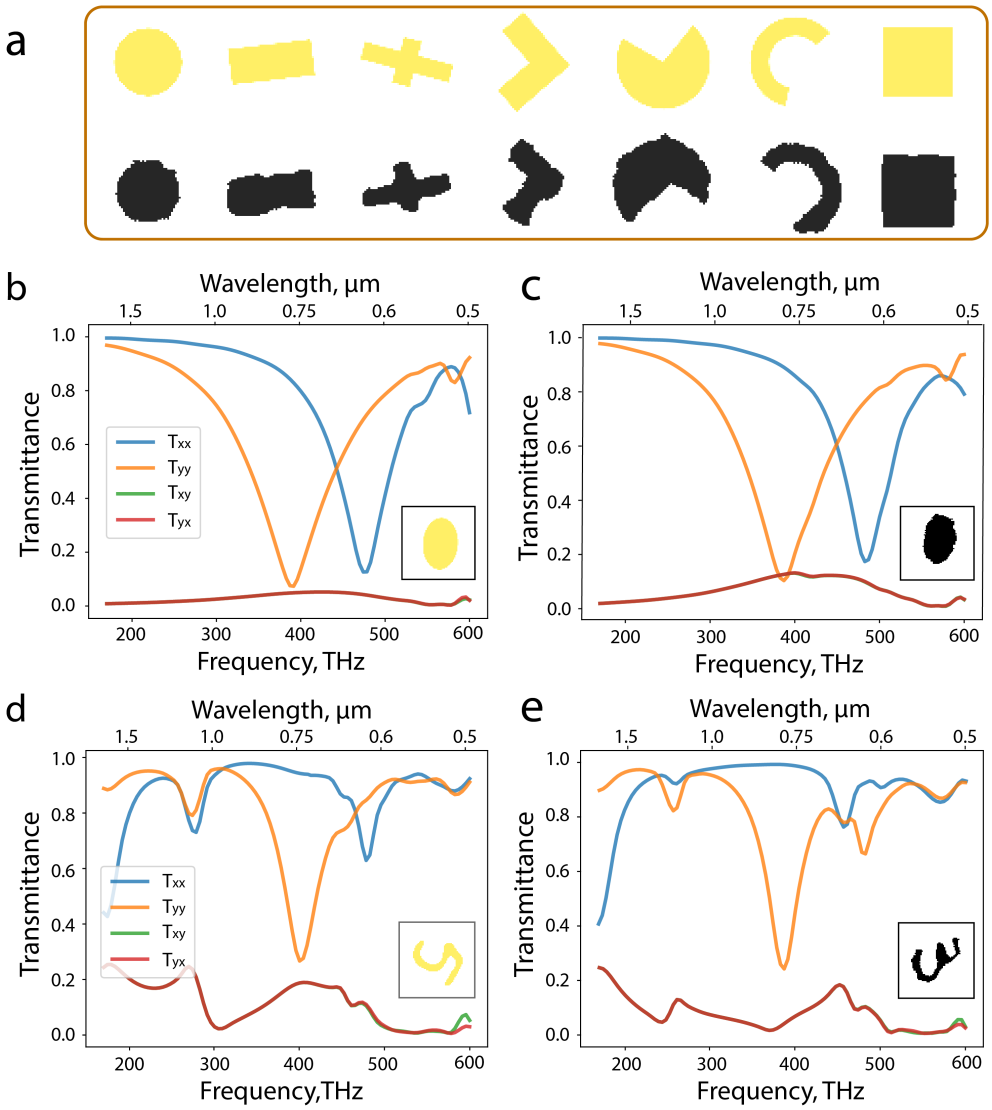


Figure 3.6: Generating patterns with a predesigned class of geometric data. (a) Test patterns s are depicted in the top row and the corresponding generated patterns s' are listed in the bottom row. Each shape provides a sample of the different classes of geometric data input to the critic network. (b) Transmittance spectra, T', of a test pattern s, to be fed to the network. (c) FEM simulated transmittance of the retrieved pattern s', from the neural network based on the input as in (b). The unit cells of s and s' are shown in the lower right corner of each figure. This result is achieved when the critic only receives geometric data of the elliptical class. (d – e) An example of results with a modified MNIST handwritten digital dataset as the input geometric data. Note that in this experiment, we intentionally excluded digit "5" in the input geometric data.

alent patterns s' that are different from the test structures s while yielding the same spectral behavior. Such examples can be found in the cross, sector and arc cases in Figure 3.6(a), where the discovered patterns s' are mirror-flipped counterparts of s with the same optical responses under linearly polarized illuminations.

Figure 3.6(b) and (c) show the spectra of an ellipse test sample s and those of the discovered pattern s', with the corresponding unit cell of the metasurface placed as insets. Comparison between the two sets allows us to conclude that the network has successfully identified the correct structure to replicate the spectra with only minor deviations. We also note that the geometric data fed into the critic network does not necessarily contain the right shape of the resultant solutions. If the type of the right pattern s for the required spectra is contained in the geometric data at the input, the nature of the GAN will lead to a decent chance of identifying the structure s as a proper candidate; otherwise, the critic will guide the generator to produce patterns with geometric features similar to those of the right geometry.

To illustrate the ability of our model to optimize structures of any shape, we further test our network with handwritten digit dataset MNIST [145] as the unit patterns of metasurfaces. We modify patterns from the MNIST dataset with rotation and shifting, and then feed them into the critic during the training. In this experiment, we intentionally exclude digit "5" in the geometric dataset at the input of the critic, and ask the network to generate a pattern that replicates the spectral features of a metasurface with digit "5". Figure 3.6(d) and (e) show the spectra of the test sample s – a rotated digit "5", and the discovered pattern s' – a modified digit "3", respectively. The topologies of the s and s' differ considerably, but the overall transmittance behaviors of the two samples, especially T_{xx} and T_{yy} , possess similar features in terms of both the spectral location and the amplitude. By removing the input structure from the trained data set, the network proves its overarching utility in producing a set of spectra that can competently match that of the input structure.

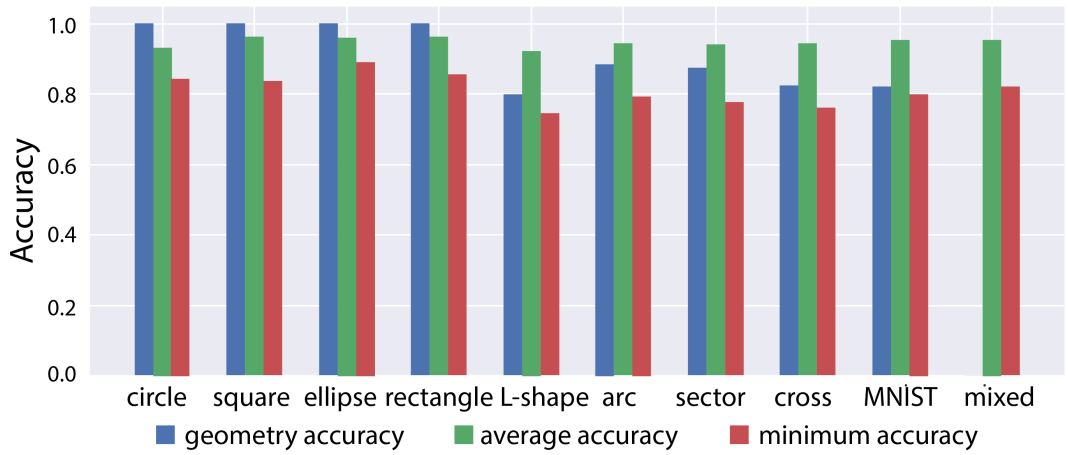


Figure 3.7: Geometric accuracy, average accuracy, and minimum accuracy of experiments with the different classes of geometric data.

3.4.3 Statistics of the design accuracy

To quantify the performance of the network, we quantitatively define three types of accuracy as follows. 1) Geometric accuracy a_{geom}: the portion of s' that can be recognized as within the same class of the input geometric data; 2) Average accuracy $a_{ave} = 1 - e_{ave}$, where e_{ave} is the average absolute error of the transmittance per frequency point; 3) Minimum accuracy $a_{min} = 1 - e_{max}$, where e_{max} represents the largest absolute error of the transmittance over all frequencies. The detailed mathematical definitions of the geometric, average, and minimum accuracies are as follows:

Geometric accuracy:

$$a_{geom} = \frac{n_g}{n} \tag{3.1}$$

where n is the total number of spectra input to the model and the n_g is the number of searched patterns that can be recognized in the same class of the geometric data fed into the critic.

Average accuracy:

$$a_{ave} = 1 - \frac{1}{f_{max} - f_{min}} \int_{f_{min}}^{f_{max}} \|T - T'\| df \tag{3.2}$$

where T is the input spectra, T' is the FEM simulated spectra of the generated patterns, and f_{max} / f_{min} is the maximum/minimum frequency of the input spectra.
Minimum accuracy:

$$a_{min} = 1 - \max_{f} \|T - T'\| \tag{3.3}$$

The geometric accuracy is used to evaluate the similarity of the generated patterns compared with the geometric data. In practice, this accuracy is not of interest if the aim of the design is to find a pattern with a spectrum the same as the input one. For fabrication and other practical purposes, this accuracy is essential if we desire to eliminate unnecessary patterns such as sharp corners and tiny dots. The average and the minimum accuracy defines the discrepancy of the input spectra T and the spectra of the generated patterns T'. Note that these two definitions do not imply the physical agreement of the input and generated spectra such as the consistency of the peak frequency and the bandwidth, thus the patterns cannot be regarded as valid only based on these two accuracies.

Figure 3.7 displays the three accuracies in our experiments for various classes of geometry. The average and minimum accuracies are calculated based on the correct ones in s' in terms of the geometric accuracy. We also note that these accuracies are data-dependent and may vary when the distribution of geometric data changes.

3.4.4 Design with all geometric data

To further exemplify the generality and versatility of the constructed network, we feed into the critic mixed data from all classes of geometry with over $8,000$ data points, and the test samples are also randomly selected from all possible geometries. As shown in the last column of Figure 3.7, the accuracy in this situation does not degrade compared to prior studies with only a single geometry input. An example of a discovered structure with mixed geometric input is presented in Figure 3.8(a) and (b). Just as before, Figure 3.8(a) and (b) indicate the FEM-simulated spectra of the test pattern s and the generated pattern s' in response to the spectral demand, respectively. Although no specific class of geom-

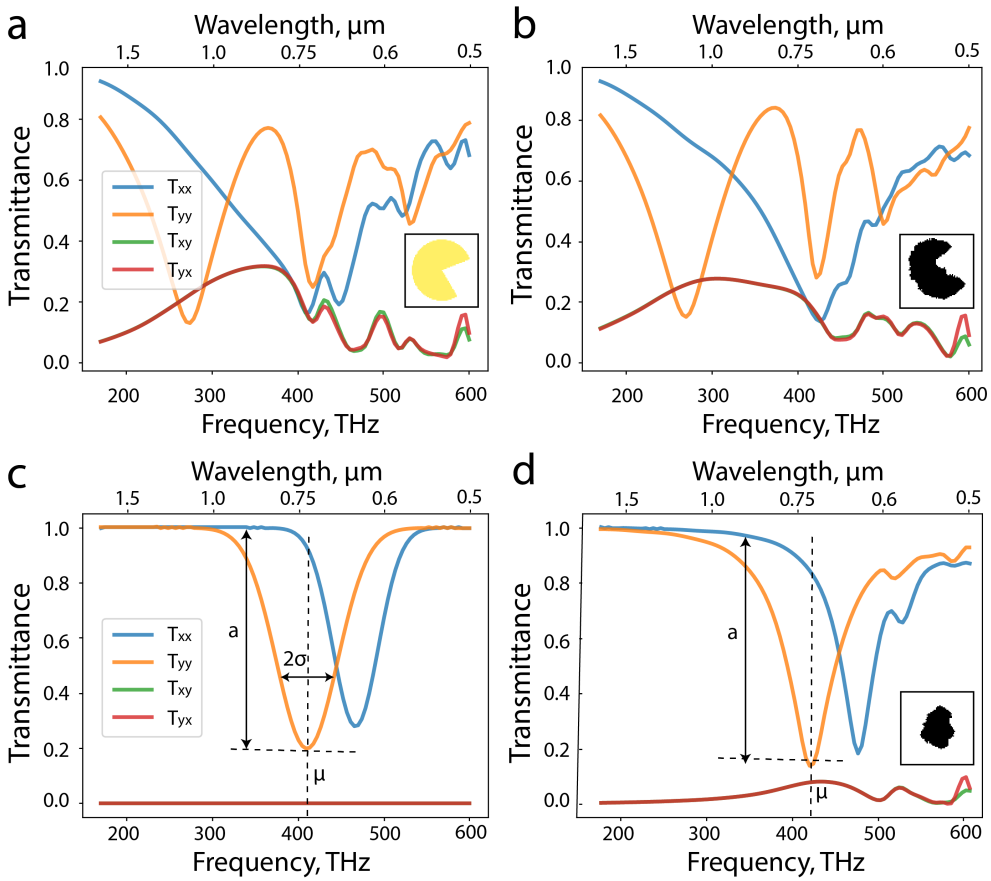


Figure 3.8: Examples of generated patterns with mixed classes of geometric data. (a – b) Example of results with a mixture of different classes of geometric data used at the input. Transmittance spectra of the test structure s and the generated pattern s' are shown in the (a) and (b), respectively, with the unit cell depicted as the inset in each figure. (c – d) Example of Inverse design of metasurfaces with human-defined spectra. (d) Desired transmittance spectra as the input to the generator, where T_{xx} and T_{yy} are two randomly generated Gaussian-like responses with parameters a, μ, and σ, while T_{xy} and T_{yx} are 0 throughout the frequency range of interest. (e) The resultant unit cell generated by our model to fit the target spectra, along with the FEM simulated transmittance spectra of this generated metasurface.

etry is indicated during the training, the generator is able to reach a pattern, s', that not only geometrically resembles s, but more importantly, possesses transmittance spectra, T', nearly identical to that of T. In general, if the input geometric data contains more than one topology that satisfies the spectral demand, the network may generate some or all of them in a probabilistic manner. Moreover, by changing the distribution of the geometric data, the user may achieve diverse solutions in response to the same spectral request, thereby mitigating the potential of degeneracy, as described earlier.

3.4.5 On-demand design examples

As a final example, we demonstrate the efficacy of our approach in the reverse design of a metasurface for user drawn spectral responses, T. In practice, the desired spectra at the input are user defined, and the existence of solutions is not guaranteed. This is particularly true when certain constraints are applied to the metasurface design. For instance, in the present study, parameters such as the materials used, the unit cell size, and the thickness of the patterned layer are all predefined in the training data. Nevertheless, when the simulator is sufficiently robust, the network is still able to unearth the best possible pattern that yields spectra T' with minimized deviation from the input spectra T. To demonstrate this feature, here we design a metasurface with the desired, user-defined spectra behavior shown in the Figure 3.8(c): i) T_{xx} and T_{yy} are two Gaussian-like resonances with randomly chosen mean μ , variance σ , and amplitude a, and ii) T_{xy} and T_{yx} are zero. The generated pattern along with its spectra T' is shown in the Figure 3.8(d). Although there exists no exact solution to spectral demand described above, the network eventually generates patterns whose spectra share common features with the input spectra including the resonance frequency, the spectral bandwidth, and the transmission magnitude.

3.5 Training dynamics of the network

3.5.1 Training of the simulator

The training of the simulator follows the common method for training a CNN. The loss of the network is defined as:

$$L_S = \frac{1}{N}\|T - T'\|^2 \tag{3.4}$$

where N is the number of the entries of T. Equation (3) defines the distance of the approximated transmittances and the input transmittances. During the training, as the total number of data is limited ($6,500$ in total), we augmented the dataset in the following ways so that it maximumly improves the performance of the simulator. First, we randomize the pixel values in the way that the regions with and without gold are represented with uniformly distributed random numbers in $(0, 1]$ and $[-1, 0]$ respectively. Second, we randomly shifted the patterns in the images with the periodic boundary conditions in both x- and y-directions. This augmentation does not require any change in the FEM simulated spectra T. Third, we randomly rotate the patterns in the images with $90°$, $180°$ or $270°$. When the patterns are rotated $90°$ or $270°$, it is necessary to exchange the T_{xx} , T_{xy} to T_{yy}, T_{yx} respectively in the loss of the simulator. Note that it is forbidden to flip the patterns, in which the case the T_{xy} and T_{yx} of some asymmetric structures cannot be exchanged. Figure 3.9(a) presents the variation of the loss during the training of the simulator.

3.5.2 Training of the generator and the critic

The generator G and the critic D themselves construct a GAN. The weights of G and D should be updated simultaneously. In our implementation, we utilized the training method

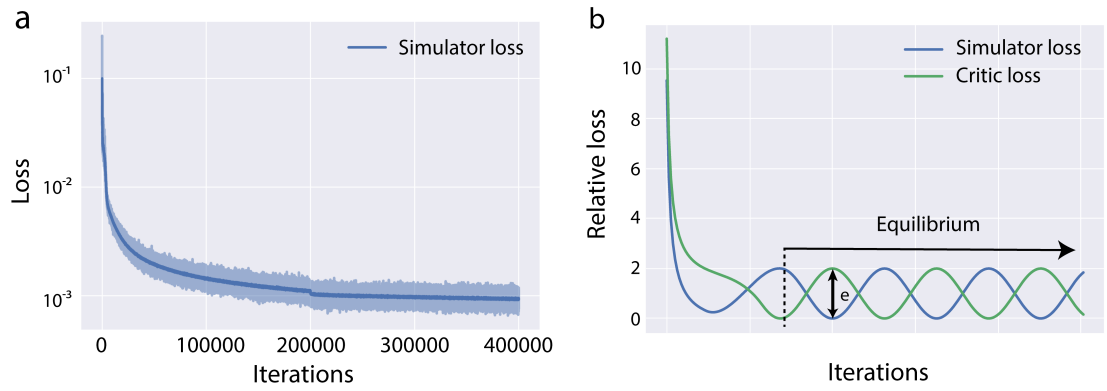


Figure 3.9: Variation of losses during the training of the simulator and the generator (a) The loss of the simulator during the pre-training of the simulator, where the lighter-colored line represents the actual loss change and the dark line is the moving average of the loss. The break point around iterations of 200,000 is caused by the change of the learning rate. (b) The general loss variation of the simulator and the critic during the training of the generator. The maximum distance e of the simulator loss and critic loss in equilibrium indicates the mismatch of the geometric data and the actual patterns given input spectra. Valid patterns are documented in this equilibrium while both and are small.

proposed in the Wasserstein GAN [146], in which the loss of the critic is defined as:

$$L_D = \frac{1}{m}\sum_{i=1}^{m} D(x^i) - \frac{1}{m}\sum_{i=1}^{m} D(G(T^i, z^i)) \tag{3.5}$$

where m is the batch size, and the superscript i indicates the ith data in the batch. The loss is to measure the distance of the two distributions formed by the geometric dataset x and the generated data $s' = G(T, z)$. Informally, this loss defines the similarity of the patterns between the geometric data and the generated patterns. We also implemented the gradient penalty introduced in Ref. [147] to improve the training performance.

The parameters of the generator are updated through back propagation subject to the losses from both the simulator and the critic:

$$L_G = L_S + \lambda L_D \tag{3.6}$$

where λ is the parameter to balance the weight of L_S and L_D . We choose $\lambda = 0.0001$ for all the experiments. The weights of the generator and the critic are updated in an alternating

order. Since the generator accepts feedback from both the simulator and the critic, it may not always result in ideal patterns as needed if there are contradictions happening in S and D. The general dynamics of the loss of the critic L_D and the simulator L_S are shown in the Figure 3.9(b). After some iterations, the losses of the two networks tend to present oscillations (equilibrium). We define the greatest distance of the two losses as e at this equilibrium stage. Since L_D defines the similarity between the generated patterns and the geometric data, and L_S defines the accuracy of the spectra of the generated patterns, we can regard e as the mismatch between the patterns s and the geometric data x. Note that as the simulator is not perfect in our experiments, sometimes it leads to the wrong approximated spectra. When the retrieved pattern has a topology too distinct compared to the training data of the simulator, the resulting structure is inaccurate. An example of this can be seen in the second column of Figure 3.5, which shows a densely distributed structure composed of small dots (similar to random noise). Therefore, satisfactory patterns are achieved in the training when L_S and L_D are both reasonably small. This is the crucial reason for having to document the possible patterns throughout the whole training process.

In different periods during the equilibrium, it is common to see various topologies occur during the training. This occurs when e is large causing discrepancies between the generated patterns and the geometric data. In our experiments, we observe that e always nears zero when the patterns of the input spectra do exist. This can also occur when the input spectra do not commonly appear in physics, which can lead to an extraordinarily large e, and the curve of the two losses will oscillate violently.

3.6 Additional test design examples

3.6.1 Additional test results with input spectra from existing metasurfaces

In this section, we show some more examples that are generated from the network with comparisons to the test patterns as in Figure 3.10. In each unit panel of Figure 3.10, the left and right figures are the FEM simulated transmittances of the test and generated patterns

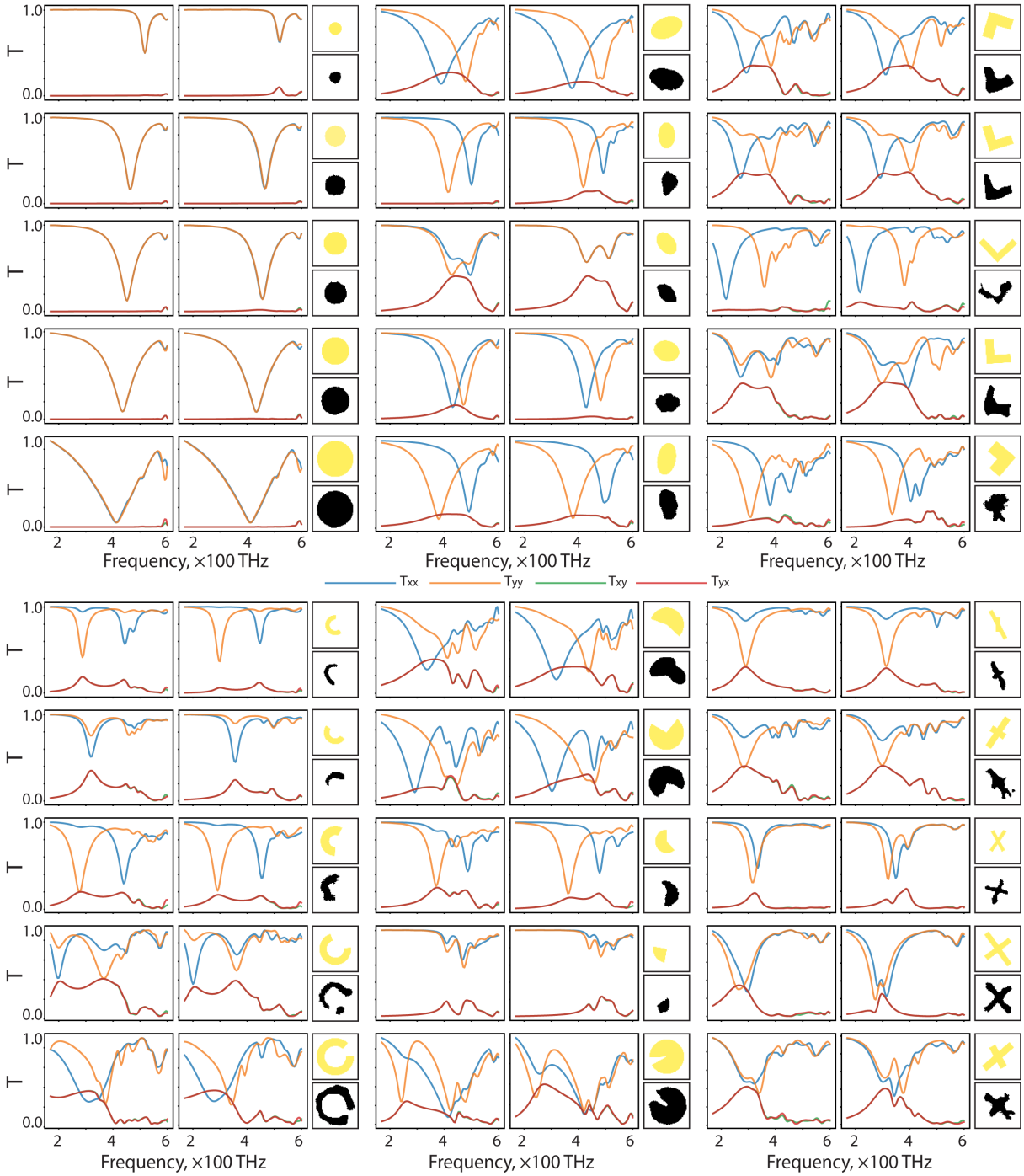


Figure 3.10: Additional patterns generated from the network with comparisons to the test patterns. In each unit panel, the left and right figures are the FEM-simulated transmittances of the test and generated patterns respectively. The corresponding unit cell of the test (yellow) and generated (black) patterns are also shown on the right. All the samples shown above are derived when a single class of geometric data is fed into the critic.

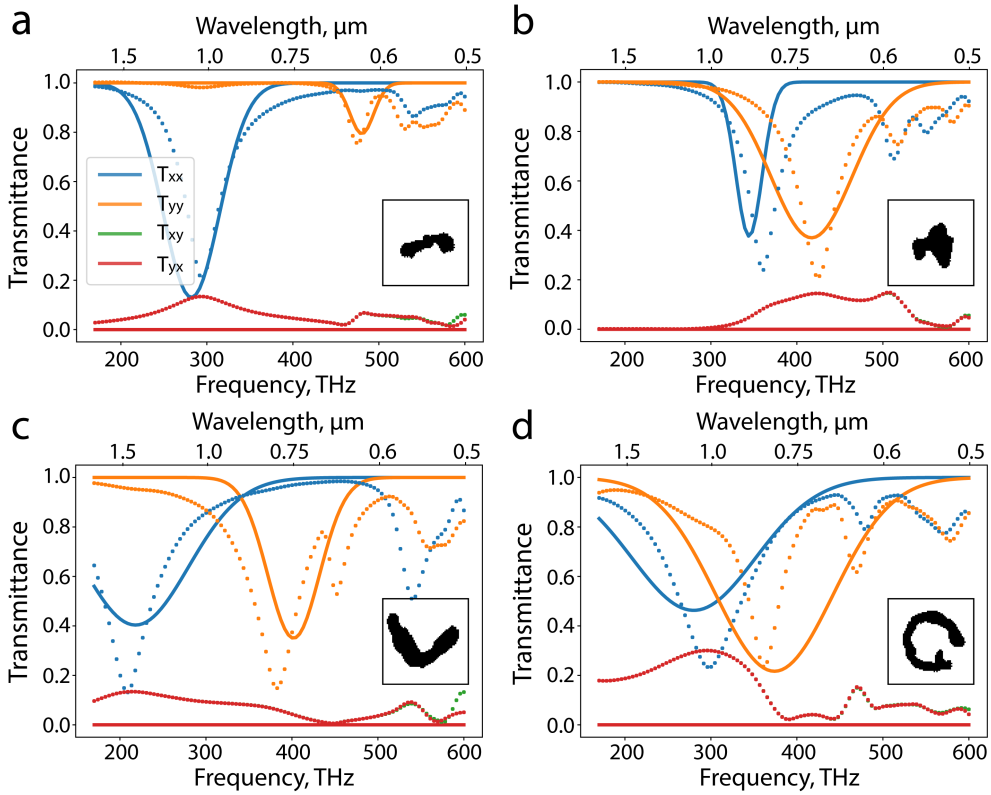


Figure 3.11: Results of the experiment with a randomly Gaussian-like input spectra. In this experiment, the input T_{xx} and T_{yy} are two randomly Gaussian-like spectra, and T_{xy} and T_{yx} are zero throughout the frequency range. The data input to the critic are a mix of all classes of geometric data.

respectively. The corresponding unit cell of the test (yellow) and generated (black) patterns are also shown on the right. All the samples shown are derived when a single class of geometric data is fed into the critic.

3.6.2 Additional test results with manually defined input spectra

Figure 3.11 displays four examples of searched patterns when T_{xx} and T_{yy} are randomly generated Gaussian-like transmittances. The pattern searched in the Figure 3.11(a) perfectly match the input spectra. However, in Figure 3.11(b) and (c), the transmittances of the searched patterns are lower than the desired input, while the peaks are in the same frequency ranges. This reflects the fact that when no patterns exist for the input spectra, the generator will compromise to produce some other patterns while minimizing the loss of the

simulator L_S . Figure 3.11(d) is an example when the broadband property of T_{xx} cannot be satisfied. Here, to minimize L_S , the network generates a pattern with two peaks residing in the frequency range of the desired bandwidth of T_{xx}. As the geometric data does not contain the exact patterns that corresponds to the input spectra, the mismatch e discussed in section 3.5.2 occurs in the equilibrium. While the losses of the critic and the simulator alternate, the patterns searched vary along with the increasing iterations. It is possible to achieve several patterns whose spectra T' have a similarly minimized distance to the input ones T.

3.6.3 Examples with partial input spectra

In practice, we may only want to search the patterns that satisfy some components of the transmittance T_{ij} for a certain frequency range. This task is easier than the ones that require all transmittances to be satisfied. This is because more corresponding patterns of the input transmittances exist when the generator receives less restricted input spectra. Figure 3.12 shows two examples of the experiments with only T_{xx} or T_{xy} input to the generator. The experiment is carried out such that the critic only receives $1,000$ arcs from the geometric data set. In Figure 3.12(a), the blue and red lines are the FEM simulated T_{xx} of the test arc pattern s and searched pattern s', respectively. The pattern of s (yellow) and the pattern of s' (black) are also presented on the right of Figure 3.12(a). s' is topologically different as s, while the T_{xx} of s' only presents a small red shift compared to the T_{xx} of s. Figure 3.12(b) shows another example with T_{xy} as the input to the generator. At this time a searched pattern s' is the same as the test pattern s, as indicated in the right part of the Figure 3.12(b). All the examples above demonstrate that our model can be used as a general method to search patterns given any transmittance component and frequency range.

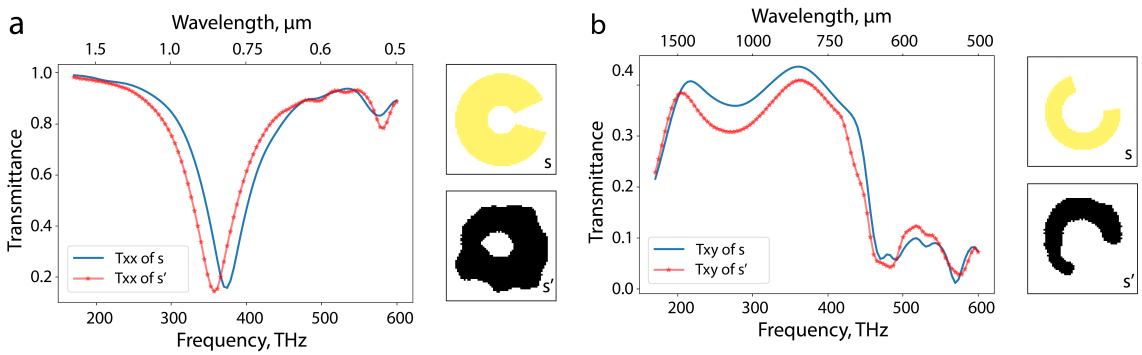


Figure 3.12: Examples of the generated patterns with partial input spectra. (a) The blue line is the test T_{xx} input to the generator and the red line with dots is the FEM simulated T_{xx} of the generated pattern s'. The unit cell of the test pattern s and the generated pattern s' are also shown in the yellow and black images on the right respectively. (b) The comparison of an example with the same settings as in (a), but the input to the generator is T_{xy} of a different pattern s, the unit cell of which is shown in the image on the right.

3.7 Discussion and conclusion

In this work, we have proposed a generative, deep, network model which efficiently discovers and optimizes unit cell patterns of metasurfaces in response to user-defined, on-demand spectra at the input. Our model addresses the pursuit of reverse design technology for photonic structures, and helps to relieve computational and specialist resources in traditional metasurface design from iterative simulations and parameter sweeping, up to generalizing the process for users lacking a solid knowledge base in optics. The model developed here is mostly based on the unsupervised learning, which guarantees efficient generation of structural patterns independent of human experience. This feature is crucial in the investigation of new structures and novel phenomena in optics and beyond. In addition, as our model can process multiple input spectra without the loss of efficiency, the workload for complicated problems that require multiple metasurfaces or gradient structural distributions will be significantly reduced. The performance of the model can be further improved by applying certain refinements, such as the use of a more sophisticated network configuration and the introduction of loss functions with more practical physical meanings. The methodology we have developed is readily extended to the pursuit of desired complex values of the reflec-

tion and transmission coefficients, which is essential to the applications like meta-lenses and meta-holograms where both the amplitude and the phase of the light waves matter. For example, by replacing our current simulator with the simulator trained to approximate the spectra and the phase delay of all-dielectric nanostructures, our approach is able to design high-performance metasurface controlling intensity and dispersion angle of light in arbitrary wavelengths. In the current work, we restrict the unit pattern of the metasurface as a single metallic particle with continuous topology. By revising the geometric dataset and refining the simulator, unit cells with multiple particles and complicated structures can be designed and optimized as well. The increased arbitrariness of topology can significantly improve the spectral accuracy of the outcome from our model. Moreover, if sufficient data for the training of the simulator is available, greater degrees of freedom in the design can be made by slightly modifying the network architecture to include modifications of the lattice constant or optimization of the pattern thickness. In the future, it is promising to incorporate additional AI algorithms, such as the reinforcement learning and the evolutionary algorithms, to reduce the dependency of the prior knowledge of human and alleviate the amount of data for the training. The developed framework can be adapted to many other applications in optics and materials science, including photonic crystals, 3D metamaterials, imaging systems, phase transitions, etc. We envision broad and growing utilization of the deep learning technology in the physics realm, so that scientists and engineers will be largely relieved from tedious processes of trial and error, and instead focus more on truly creative thoughts yet to be broached by the machine.

CHAPTER 4

COMBINE DEEP LEARNING WITH EVOLUTIONARY ALGORITHMS

To automate the inverse design of optical structures with minimal intervention of human, and to discover arbitrarily-shaped photonic building blocks with minimal predefined restrictions, here we introduce a fast, robust, and generalizable strategy, consolidating a deep generative model and a modified evolution strategy (ES). The designing process only takes customer-defined, on-demand physical properties as the input, without any other prior knowledge. Moreover, the framework is capable of generating photonic patterns with either continuous or discrete topology. A generative model, specifically a VAE in this case, is exploited to encode all potential structures into a latent space, and the modified ES algorithm is applied to the latent vectors to identify an optimum pattern whose physical properties fit the target ones at the input with minimized discrepancy.

4.1 Generative model as an encoding method for photonic structures

4.1.1 Encoding photonic structures with GAN and VAE

Globally optimizing the topology of photonic structures represented by binary images is a sought-after goal. However, the high DOF of the image space impede an effective scheme for the topology optimization. We hereby utilize deep generative models to encode the binary images into a continuous, low-dimensional latent space, and in doing so, the optimization can be applied on the latent space to effectively and expeditiously identify the topology of the pattern in a global manner. GAN and VAE are two basic yet fundamental deep generative models. Although the architectures and training methods of the two networks are distinct, both include a latent space and a generator/decoder. The trained generators of the VAE and the GAN can recover data that are similar to the training dataset

from the latent space. In our framework, the GAN and the VAE are equivalent in terms of effectiveness and functionality. Here the VAE was chosen for the compression of topology of photonic structures.

4.1.2 Training of the generator in a VAE

Figure 4.1(a) represents an architecture of a vanilla VAE used to reduce the dimensionality of the topology of photonic structures. The encoder transforms the input geometric data s represented by binary images into mean vectors μ and standard deviation vectors σ. During the training process, latent vectors v are sampled from the Gaussian distribution $v \sim N(\mu, \sigma)$. The decoder G then reconstructs v back to the geometric information. In doing so, pattern topologies with similar features are mapped to the same region in the latent space, so that the decoded patterns with similar features can be continuously varied by perturbing the latent vectors. Informally, the objective of the training is to optimize the weights of the decoder G so that the randomly sampled v can be mapped to topologies that are similar to the training dataset. To enforce global continuity of the latent space, additional regularization is needed during the training of the VAE. This is achieved through minimizing the Kullback-Leibler (KL) divergence between the distribution of the latent vectors $q_s(v)$ and a standard normal distribution $p(v)$. In detail, the loss function of the training is as follows:

$$L = \mathbb{E}_{v \sim q_s}[\|G(v) - \hat{s}\|] + \beta \mathrm{KL}(q_s(v), p(v)) \tag{4.1}$$

where $\hat{s}$ is the reconstructed pattern from the decoder. The first term is the reconstruction loss controlling the similarity of the generated patterns and the training dataset, and the second term is the KL divergence enforcing the distribution of the latent vectors. Here β is a weight to balance the KL regularization and the reconstruction loss. In our experiment we set $\beta = 1.5 \times 10^{-4}$. Reparameterization is utilized to avoid the block of gradient caused by the Gaussian sampling during the training.

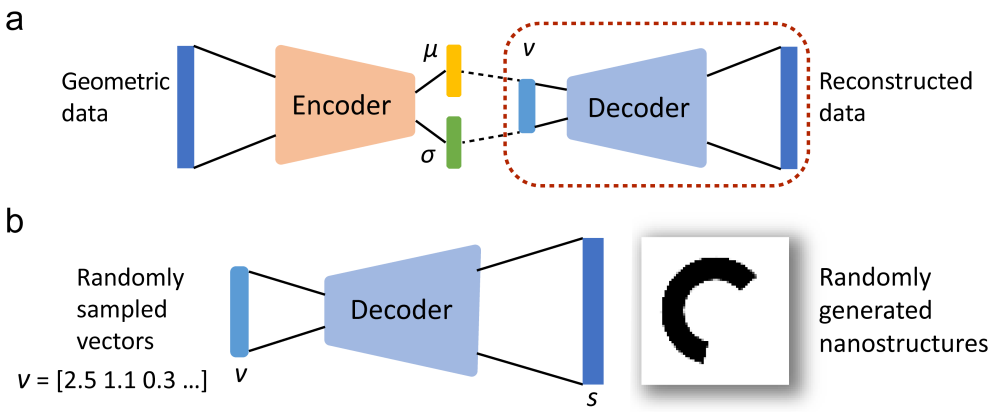


Figure 4.1: Basic architecture of a VAE as an encoding method for photonic structures (a) The illustration of the vanilla VAE. The encoder accepts the geometric data and produces two parametric vectors μ and σ. Random vectors v are samples from the normal distribution with the mean and the standard deviation defined by μ and σ. The decoder then reconstructs vectors v to images of photonic structures. The VAE encodes the geometric data into a compact latent space where the optimization algorithms can be applied efficiently. (b) After the training, the decoder encircled in (a) can be separated and treated as a generator of geometric data. It transforms randomly sampled vectors v to their correspondent structures.

After the training of a VAE, the decoder of which can be operated as a geometric data generator as indicated in Figure 4.1(b). When fed with a randomly sampled vector v in the latent space, the corresponding pattern of the structure s can be reconstructed. Optimization algorithms can be applied to the latent vector instead of the image representation to substantially reduce the dimensionality of the design problem. We note that an autoencoder (AE) can also be used for topological dimensionality reduction. However, because an AE simply transforms a high-dimensional image into a sparse representation without considering the continuity of the latent space, optimizing the latent vector of the AE is not as effective as the GAN and VAE optimization.

4.2 Identifying optimal structures with evolution strategy

As a result of the reduced dimension of the sparse representation enabled by the generative model, we can apply optimization techniques on the low-dimensional latent space, instead of having high-dimensional binary images representing the topology of structures, to ex-

peditiously identify the structures with certain input objectives. Here we implement the ES with modification to globally search the optimum latent vector v given certain design objectives q. Figure 4.2 illustrates the flowchart of the framework consolidating both deep generative models and the ES. In the implementation process, a population of all individuals is first initialized. Each individual contains two random vectors – a latent vector v and a mutation strength m. The mutation strength m is a vector with the same dimension as v and will be used to randomly vary the value of v in a later step. After initialization, all individuals are reconstructed to their corresponding patterns s through the generator G. The recovered structures are then simulated through either a physical simulator or a neural network approximator. In order to avoid repeated simulations and expedite the optimization process, we adopt a neural network (S) trained with simulated data to predict the physical responses $\hat{q} = S(s)$ of the structures. With the design objective q and the simulated result $\hat{q}$, we calculate a fitness score r, representing the agreement of q and $\hat{q}$, for each individual. The value of the fitness score is calculated through a fitness function $r = F(\hat{q}, q)$, where the form of F is designed by the user based on the goal of the optimization. For example, if q is a vector representing the spectral responses, the fitness function can be defined as $F(\hat{q}, q) = \|q - \hat{q}\|$. Since the ES is a black-box optimization algorithm, the fitness function can be a non-differentiable function. The arbitrariness of the fitness function augments the applicability of the algorithm in designing photonic structures with complex design objectives.

After the evaluation of fitness scores r, the algorithm enters the optimization stage that emulates natural selection. In detail, μ best individuals with the highest scores are selected as parents (P), and their reproduction leads to the next generation of children (Q) with a total number of individuals λ. A latent vector v_{child} of a child is created from two parent vectors v_1 and v_2 via two manners, 1) crossover, which randomly exchanges segments of vectors of the two parents, and 2) interpolation, which linearly interpolates the vectors of the two parents $v_{child} = \alpha v_1 + (1 - \alpha)v_2$ with random weights $\alpha \in [0, 1]$. The mutation

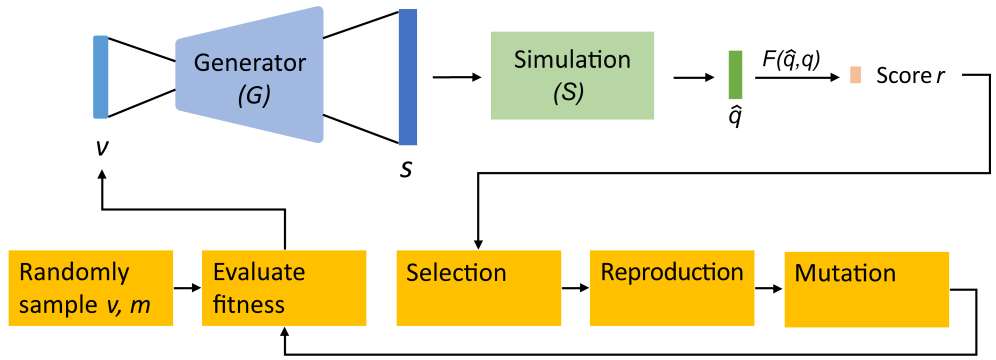


Figure 4.2: Flowchart of the VAE-ES framework. The developed approach employs a traditional ES algorithm with a loop of evaluation of fitness, selection, reproduction, and mutation.

strength of a child m_{child} is derived in the same manner as v_{child}.

The λ reproduced individuals together with the μ parents constructs the next generation, which is known as the $(\lambda+\mu)$ strategy. In the mutation stage, noises sampled from the normal distributions $\delta v = N(0, \mu)$ are added to the latent vector $v+\delta v$ so as to avoid stagnancy of the evolution. The mutated population forms the next generation of the population in the optimization process. The whole algorithm then performs the iterative process of reconstruction, simulation, evaluation, selection, reproduction, and mutation until a satisfactory individual with an optimized score is identified. During the optimization, we occasionally remove individuals v that stay unchanged for a few iterations to reduce the unnecessarily repeated simulation and speed up the convergence. The algorithm stops once the design criteria are achieved or the maximum iteration is reached. The detailed algorithm is shown in Algorithm 1, where the function Reproduce and Mutate are as described above. We hereby name the framework the VAE-ES framework, as it represents a consolidation of two separate algorithms, the VAE and the ES.

Although we implement the ES as the optimization algorithm, we note that other derivative-free optimization techniques such as random search, Bayesian optimization, and other global/local optimization algorithms can be used to replace the ES in the framework. Such replacement is considered when, for example, intensive simulations are inevitable or local optimization is sufficient. Our framework provides a flexible approach to accommodate the

optimization of structures with various requirements and objectives.

Algorithm 1: The hybrid framework of VAE-ES

Input: design objective q, number of individuals in the population N_{ind}, fitness function F, elite number M, and latent vector dimension d

Output: optimized pattern s_{opt}

initialization: generator G, simulator S, latent vectors and mutation strengths $v, m \in \mathbb{R}^{N_{ind} \times d}$, optimal vector $v_{opt} \in \mathbb{R}^{d}$, fitness scores $r \in \mathbb{R}^{N_{ind}}$ of all individuals, scores of the optimal vector $r_{opt} = -inf$;

while *NOT one of the individuals reach criteria* **do**

Decode latent vectors to structure patterns $s \longleftarrow G(v)$;

Simulate the decoded structures $\hat{q} \longleftarrow S(s)$;

Calculate the fitness scores of the population $r \longleftarrow F(q, \hat{q})$;

if $\max(r) > r_{opt}$ **then**

$j \longleftarrow \text{argmax}(r)$;

$r_{opt}, v_{opt} \longleftarrow r[j], v[j]$;

end

Select latent vectors and mutation strengths v^{top}, $m^{top} \in \mathbb{R}^{M \times d}$ of top M species based on their fitness scores ; Reproduce new generation of latent vectors $v, m \longleftarrow$ **Reproduce**$(v^{top}, m^{top}, N_{ind})$;

Mutate the new generation using the mutation strengths $v \longleftarrow$ **Mutate**(v, m) ;

end

return $s_{opt} = G(v_{opt})$

4.3 Validation of the framework with metasurface design

4.3.1 Configuration of the metasurfaces

To evaluate the performance of the proposed strategy for the inverse design of photonic structures, the VAE-ES framework is applied to the design of the unit-cell pattern s of a

metasurface that corresponds to customer-defined optical spectra T fed at the input. Gold is chosen to form the single-layered nanostructure, which situates on a glass substrate. The default lattice period and the thickness of the patterned layer are set to be $380\,\mathrm{nm}$ and $50\,\mathrm{nm}$, respectively.

4.3.2 Network training details

A VAE is trained with a pyramid convolutional network architecture using a geometric dataset that contains more than $10,000$ random patterns of various shapes. The classes of all geometric data include, but not limited to, ellipse, rectangle, polygon, cross, random union/intersection of other classes, and false reconstructed samples by-produced from the training process of the VAE. Each geometric data point is represented as a binary pixel image with a size of 64 by 64, in which 1 stands for the constituent material and 0 for the air, and the encoded vector v of the images by the VAE has the dimension of 10.

To accelerate the searching speed, we implement a neural network simulator to approximate the spectral behavior of the metasurface with a specific unit-cell pattern. The simulator accepts the image of a photonic structure and approximates the four components of the transmittance T_{ij} of the metasurface, where i and j indicate the polarization directions for the incidence and the detection, respectively. The frequency range of spectra under investigation is set to be from $170\,\mathrm{THz}$ to $600\,\mathrm{THz}$, which corresponds to $500\,\mathrm{nm}$ – $1.8\,\mu\mathrm{m}$ in the wavelength domain. Constrained by the generator and limited by the fabrication requirements, the designed structures with reasonable geometric features, such as sufficient connectiveness and smoothness, should lie in small regions in the whole image space. As such, we do not necessarily need to train a complete simulator that can accurately predict every pattern that can be enumerated in the 64×64 image. Although such a simulator cannot replace a rigorous physical simulation for general purpose of simulation, it is sufficient for the design of photonic structures with high fidelity within our framework. It is worth noting that the neural network simulator used here is developed to expedite the computing

process, and can be replaced by any physical simulation method such as the FEM and the FDTD.

4.3.3 Fitness Functions and Hyperparameters of the Framework

We define the fitness function and illustrate the experiment settings. We denote the patterns identified from the framework as s', the spectra approximated by the neural network simulator as $\hat{T}$, and the transmittance of s' computed by the FEM simulation as T'. As the objective is to minimize the L_2 and/or L_∞ norm of the difference between the input spectra and the approximated transmittance, a reasonable choice of the fitness function can be written as:

$$F(T, \hat{T}) = -\|T - \hat{T}\|_2 - \beta \max(|T - \hat{T}|) \tag{4.2}$$

where β is the weight balancing the two norms. Both norms are included to stabilize the searching process and to enforce similar numerical features between $\hat{T}$ and T. Setting $\beta = N$, where N is the dimension of the vector T, allows the values of the two norms to be on the same order of magnitude. Since we do not rely on the backpropagation algorithm in the ES, the fitness function can be non-differentiable.

In the experiment, the population size is set to 50, and the children size of each new generation is from 50 to 100. Only individuals with top 25 scores are selected as parents of the next generation. The maximum iterations of the algorithms are set to 200, which on average takes 5 seconds for each set of the input spectra with the acceleration of the simulator. If the framework achieves the desired fitness score within the maximum iterations, the optimization ends instantly. Otherwise, if the fitness score keeps under the requirement without change for 50 iterations or the maximum iteration is reached, we manually stop the optimization and determine the current solution to be optimized. Relying on the fast optimization speed and stochasticity of the ES, we can perform multiple optimizations if one solution cannot be confirmed as the global optimum.

Based on our extensive testing, most of the searching practice will stop as an optimized score is reached within 50 generations. It should also be noted that the parent size and the children size are not restrictive to the values we described above. Changing the parameters of the ES causes a very minor impact on the final performance of the design. Average accuracy a_{ave} defined in Equation 3.2 is used to quantitatively evaluate the performance of our approach.

4.4 Design results and analysis

4.4.1 Validation with test dataset

We demonstrate the capability and accuracy of our framework by applying it to the inverse design of photonic structures based on the input spectra of randomly selected, actual metasurfaces. This ensures that a reasonable solution must exist for any given input spectra of this kind. In our experiment, 500 random test patterns of metasurfaces s from all classes of geometry are selected, and FEM simulations are performed to obtain the spectra T of these metasurfaces, which are subsequently fed to the framework as the target spectra. Our algorithm is tested by identifying the corresponding pattern s' for the input spectra T and evaluating the average accuracy of the simulated spectra of s'. Statistically, the expected $a(\hat{T}, T)$ and $a(T', T)$ of identified s' are above 98% and 95%, respectively, and the difference stems from the error associated.

Figure 4.3 presents six examples of the input spectra T (solid lines) and FEM simulated spectra T' (dashed lines) in conjunction with the corresponding patterns s and s' in the unit cell. Our approach captures the prominent features of the input transmittance and generates structures with minimal discrepancy between T and T'. As the ES algorithm only considers the distance between T and $\hat{T}$ as the optimization criterion, any shapes, similar to or distinct from s, can be identified as an optimal solution only if the score $F(T, \hat{T})$ is maximized. This feature also allows us to unearth different investigate the potential of our framework with the design of structures with fine features, we present another six designs with complicated

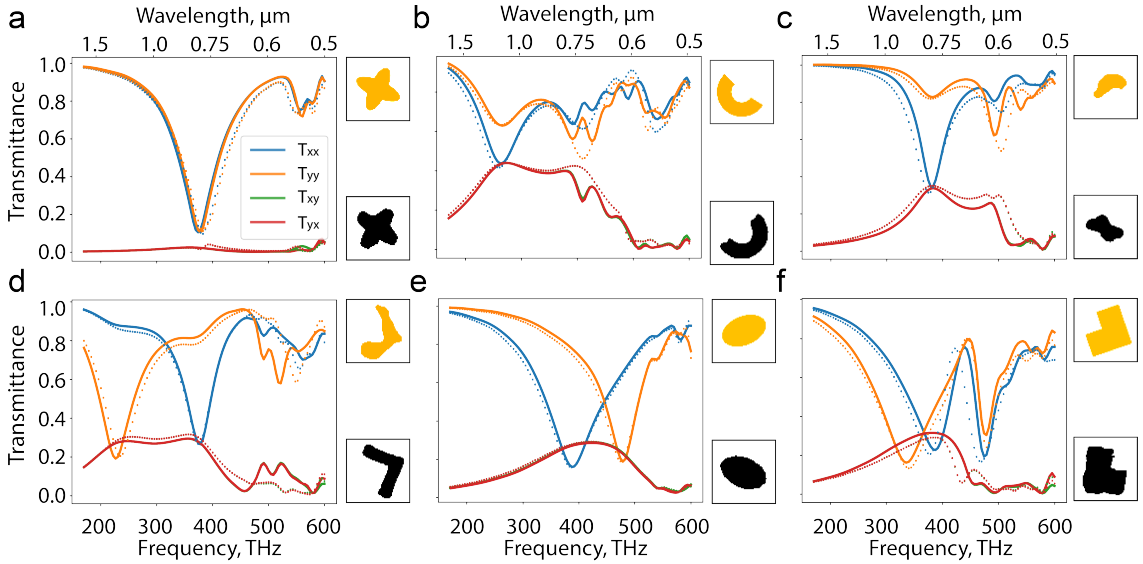


Figure 4.3: Test results of designed photonic structures with simple topologies. (a) – (f) The desired spectra T_{xx}, T_{yy}, T_{xy} and T_{yx} drawn from a series of test patterns are shown in solid lines, while the FEM simulated spectra T' of each generated pattern are shown in dashed lines. The test patterns s and discovered patterns s' are presented on the right of each figure in orange and black, respectively.

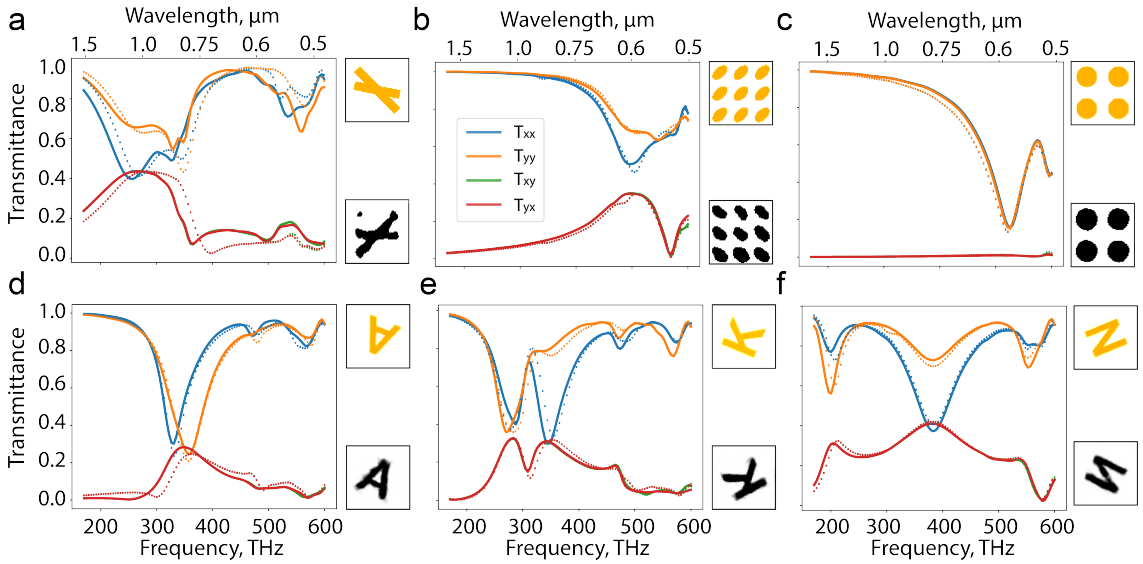


Figure 4.4: Test results of designed photonic structures with fine structures and complex topologies. (a) – (f) The desired spectra T_{xx}, T_{yy}, T_{xy} and T_{yx} drawn from a series of test patterns are shown in solid lines, while the FEM simulated spectra T' of each generated pattern are shown in dashed lines. The test patterns s and discovered patterns s' are presented on the right of each figure in orange and black, respectively.

topologies in Figure 4.4. Figure 4.4(a) – (c) display the designs with sharp corners and discrete topologies, and Figure 4.4(d) – (f) present structures of the alphabet A, K and N. Although these patterns have elaborate and complex features, our framework is able to identify the target patterns with marginal disagreement.

4.4.2 Validation with on-demand input spectra

In the next part, we will reveal the capacity of the proposed framework for the inverse design of photonic structures, based on arbitrary, user-defined input spectra. Two sets of experiments for such on-demand design have been performed, and the results are shown in Figure 4.5 and Figure 4.6. In Figure 4.5(a) – (h), the input spectra T_{xx} and T_{yy} are set to be two randomly chosen, Gaussian-shaped curves, and T_{xy} and T_{yx} are set to zero throughout the frequency range of interest. The results of the FEM-simulated spectra of the identified patterns (dashed lines) faithfully match the input spectra in terms of both the spectral location and the bandwidth. It is worth noting that the independent manipulation of T_{xx}, T_{yy}, T_{xy} and T_{yx} is not readily achievable by conventional human design approaches, while our framework is able to accomplish simultaneous control of T_{xx}, T_{yy} and suppression of T_{xy}, T_{yx}. In the on-demand design cases, we note the accuracies (indicated in the caption of Figure 4.5 cannot achieve 95% as in the test/validation cases. This is because the input spectra may not have exact corresponding photonic structures. As a heuristic algorithm, ES cannot always converge to global minima and better solutions might exist. However, as a global optimization algorithm, ES is able to avoid local minima unlike local optimizations and identify a globally optimal structure with significantly higher likelihood. On the other hand, relying on the stochasticity of the ES, a global optimum can be confirmed through multiple runs of the optimization with random initialization.

In Figure 4.6(a) – (h), we set the input T_{xx} as a notch filter without any specifications in regard to the T_{yy}, T_{xy} and T_{yx} components. We note that within the family of single-layered metallic metasurfaces, it is not possible to identify a metasurface that perfectly

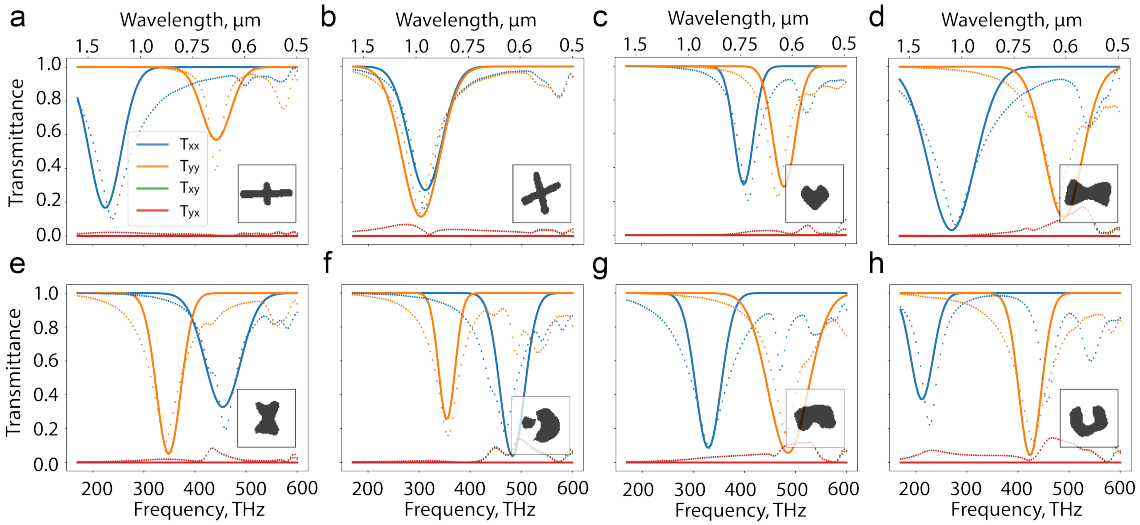


Figure 4.5: Samples of on-demand inverse design with Gaussian-like target spectra. (a) – (h) The desired spectra T_{xx} and T_{yy} at the input, shown as the solid lines, are two randomly generated Gaussian-like curves, and the T_{xy} and T_{yx} are zeros across the frequency range of interest. The generated patterns in the unit cell are depicted in the lower right corner of each panel, and the FEM-simulated spectra of the resultant nanostructures are represented in dashed lines. The accuracies of samples (a) – (h) are 0.913, 0.965, 0.933, 0.910, 0.932, 0.915, 0.875, 0.890, respectively.

replicates ultrasharp spectral features such as the steep cut-off slopes. However, the VAE-ES framework is able to generate a structure that accurately replicates the primary features of the desired filter, including the central frequency and bandwidth of the stop-band. In the general case when no pattern exists in correspondence to a given desired spectra at the input, the framework is guaranteed to generate a structure with minimized discrepancy between its spectra and the input ones. We also note that although the transmittance spectra are set as the design objective in these experiments, any photonic responses such as the diffraction behavior, optical chirality, and field localization can be used as the intended design criteria without further adjustment of our framework.

4.5 Generalization of the VAE-ES framework

As shown from the examples presented above, the VAE-ES developed here is capable of generating patterns of single-layered metasurfaces with prescribed properties, but certain

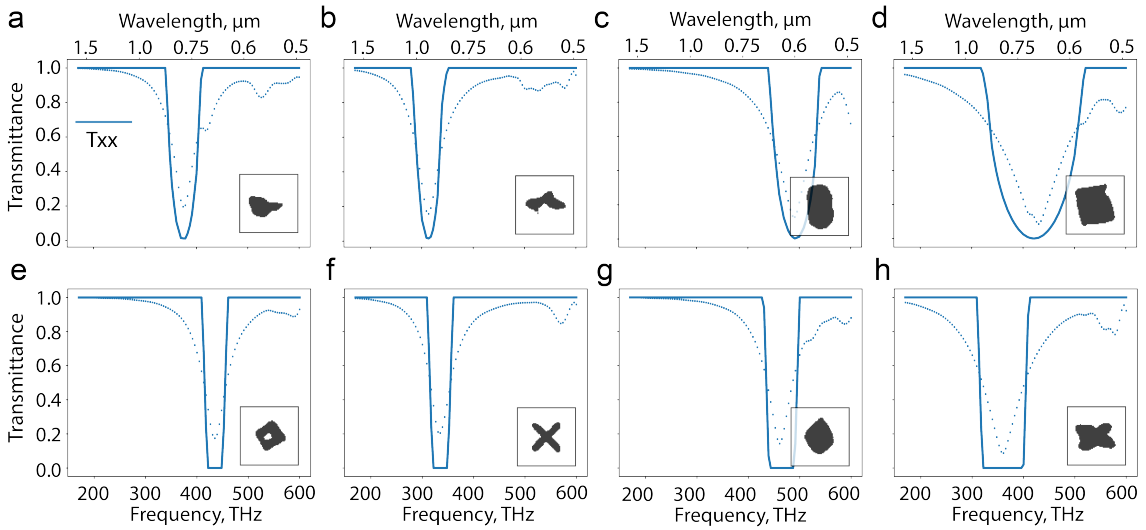


Figure 4.6: Samples of on-demand inverse design of notch filters. The desired spectrum T_{xx} at the input has a band-stop transmission feature with specific central frequency and bandwidth. All these examples demonstrate the effectiveness of the framework, which is able to generate nanostructures that resemble the on-demand spectra fed at the input, and faithfully replicate major features in terms of the spectral location and the bandwidth.

dimensional and physical parameters are preset. While limited by the availability of data, here we briefly comment on how the proposed methodology can be adapted to more complicated photonic structures that consist of multi-layered or three-dimensional build blocks. For example, by expanding the latent vectors with additional parameters, our VAE-ES approach can be applied to the optimization of an extended parameter space without any adjustment. As presented in Figure 4.7(a), when the latent vector is expanded to include extra parameters such as the thickness, the lattice period, and the materials, the VAE-ES framework will be able to identify the structure pattern while concurrently seeking the optimum parameters. Figure 4.7(b) illustrates an encoding method for the inverse design of multi-layered metasurfaces. The encoded vectors of the layers are concatenated, with an additional parameter representing the distance between adjacent layers. There are other effective encoding methods for more complicated photonic structures, especially when geometrical features along all dimensions are involved, although a complete discussion of these schemes would be beyond the scope of the current work. The common characteristic of these encoding methods is to exploit a well-trained VAE or GAN to offer a compact

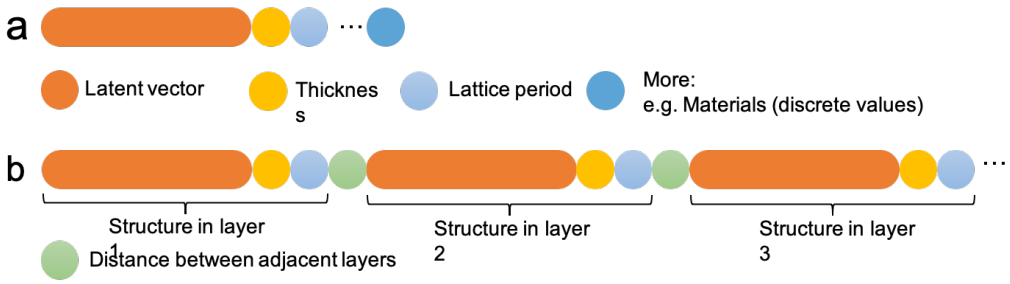


Figure 4.7: Encoding methods for the general inverse design of metasurfaces. (a) Encoding a photonic structure in a unit cell with additional parameters, such as the thickness of the patterned layer, the lattice constant of the unit cell, and the constituent materials encoded in discrete values. (b) An encoding method for multi-layered metasurfaces. A series of encoded vectors, representing photonic structures in different layers, can be concatenated as a new vector for the inverse design of multi-layered metasurfaces. The distance between adjacent layers can also be included as a parameter to be optimized.

representation of all possible candidate structures.

4.6 Discussion and conclusion

In summation, we have demonstrated a robust, efficient, and generalizable framework consolidating the evolution strategy and a generative model to automate the inverse design of photonic structures for user-defined design objectives. As a proof of concept, the capacity of the proposed framework is tested through design practices of metasurfaces in response to on-demand spectral properties. With a change of the fitness function, the framework can be adapted for the design of metasurfaces with the manipulation of other properties of light, including phase, polarization, and dispersion. Without prior knowledge of the geometry of the candidate patterns, the VAE-ES framework automatically and globally identifies the optimal structure, in continuous or discrete topology, of the metasurface within short period of time. Moreover, our approach allows flexible substitution of constituent modules in the algorithm. For example, the user can choose to replace the generator with either a VAE or a GAN trained on the dataset of interest, and determine the favorite simulation method such as a neural network (to expedite the search speed) or the FEM (to improve the reliability of the design). Unlike other neural network (NN)-based approaches, the fitness function

in our framework is not restricted to differentiable functions. As such, the fitness function can be formulated to possess a large degree of freedom to accommodate arbitrary physical objectives, an important feature shared among traditional optimization methods.

We envision that the performance of the VAE-ES framework can be consistently boosted by advanced generative models and evolutionary algorithms. Thanks to the flexible formulation of the fitness function and the scalability of the encoding method for photonic structures, the strategy developed in this work can be adapted for the design and implementation of complex photonic materials and devices for prescribed physical properties and diverse wave manipulation capacity. Without any working knowledge of optics and electromagnetic waves, our approach can be readily implemented to speed up the designing process of large-scale devices such as digital metamaterials and meta-holograms. In addition, by consolidating adjoint methods, including the coupling of gradient-based optimization and state-of-the-art generative models, we expect the development of end-to-end global design strategy for photonic structures with high performance and extensive applications. Finally, beyond the field of optics, photonics and electromagnetic waves, we envisage a generic utilization of our framework in other disciplines where people can be, to a certain extent, spared from conventional trial-and-error practices when designing complex material systems.

CHAPTER 5

ALGORITHMIC DESIGN OF MULTI-ELEMENT METASURFACES: COMPOUNDING META-ATOMS INTO META-MOLECULES

Molecules composed of atoms exhibit properties not inherent to their constituent atoms. Similarly, meta-molecules consisting of multiple meta-atoms possess emerging features that the meta-atoms themselves do not possess. Metasurfaces composed of meta-molecules with spatially variant building blocks, such as gradient metasurfaces, are drawing substantial attention due to their unconventional controllability of the amplitude, phase, and frequency of light. However, the intricate mechanisms and the large degrees of freedom of the multi-element systems impede an effective strategy for the design and optimization of meta-molecules. Here, we propose a hybrid artificial intelligence-based framework consolidating compositional pattern-producing networks and cooperative coevolution to resolve the inverse design of meta-molecules in metasurfaces. Despite the development of techniques for the optimization of photonic structures, the inverse design of metasurfaces with meta-molecules, of which the degrees of freedom are astronomical, is still not resolved. Although the collective properties of meta-molecules can be predicted by individually simulating each meta-atom, the enormous number of possible combinations of candidate structures impedes efficient designing using state-of-the-art optimization techniques. In order to effectively discover and design functional meta-molecules consisting of spatially variant meta-atoms, in this paper, we propose a hybrid framework of a CPPN and a CC algorithm.

5.1 Controlling topology of photonic structures with CPPN

Unlike convolutional neural networks (CNNs) that generate an entire image in one shot, CPPN take the coordinates of the pixels in the image as an input one at a time, predict the corresponding pixel values, and, after iterating all the coordinates, assemble the predicted

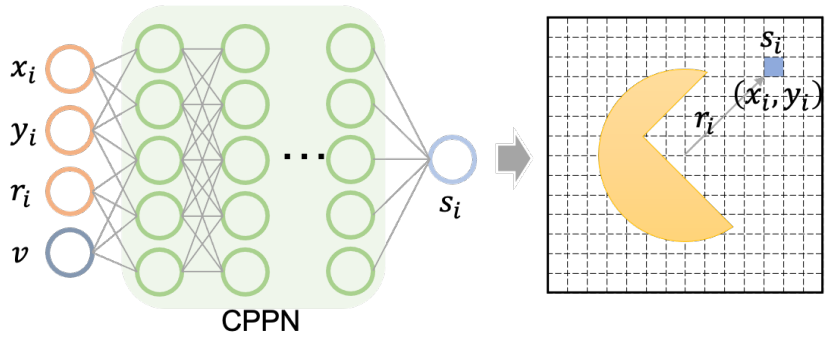


Figure 5.1: The schematic of a CPPN. The network takes coordinates of a pixelated image (x_i, y_i, r_i) and a bias vector v as the input, and predicts the value s_i of each pixel through the fully connected neural network.

pixels as a pattern [148]. As illustrated in Figure 5.1, the inputs to the network are the coordinate tuple (x, y, r) and the bias vector v, where r represents the distance of pixel (x, y) from the center of the image and v is a parameter controlling the shapes of generated patterns. Here v can be treated as the latent vector of the generated pattern since it is a low-dimensional vector representation of the high-dimensional composed image. Denoting the network as a function f and the generated pattern as s, the network essentially performs the transform $s_i = f(x_i, y_i, r_i, v)$, where the subscript i is the index of a pixel in the image. As the trained CPPN composes patterns in a pixel-by-pixel manner, it is able to generate patterns with arbitrarily complicated features including but not limited to corners, curves, and slits as desired by users. These fine features are crucial to strong resonance and coupling, and thus are indispensable to the discovery of novel photonic structures. We note that, given the geometric dataset containing patterns with proper geometric features, CPPNs are also able to compose patterns in a more randomized fashion for other inverse design problems. Implementation of a CPPN f can be realized by any neural networks, and here we choose fully connected networks for simplicity. In the following research, a CPPN is trained in the framework of GAN, as in Chapter 3, to encode the complex photonic structures and generate arbitrary topologies.

The geometries of patterns are encoded as latent vectors v's by the CPPN generator. In order to augment the generated patterns, we construct a new vector v_{cat} concatenating

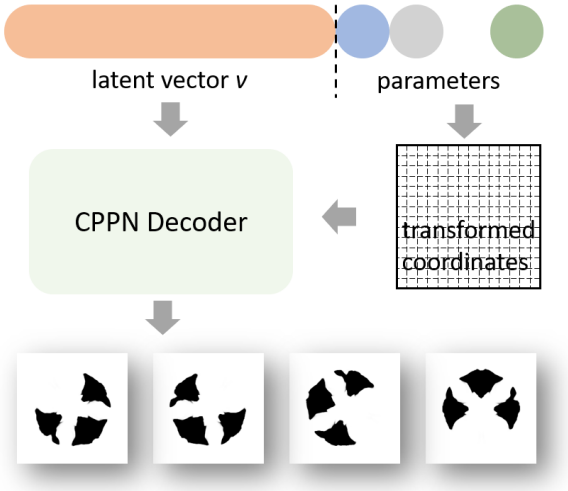


Figure 5.2: Encoding strategy in our framework. The patterns are encoded into a concatenation of the latent vector v and additional parameters. Reconstructing the meta-atoms is divided into two steps: we first perform the transformations defined by the additional parameters to the input coordinates, and then feed the transformed coordinates and the latent vector v into the CPPN network.

additional parameters that defines the transformations on the generated patterns as shown in Figure 5.2. Two strategies can be used to decode the concatenated vectors: 1) decode the v through the CPPN first and perform the transformations or 2) perform the transformation on the coordinates fed to the CPPN and directly retrieve the transformed patterns. We choose the latter one because the gradient of the patterns with respect to the concatenated vectors can be kept. In our work, we only choose to rotate the generated patterns. Therefore, we attach an additional parameter describing the rotation angle of the patterns. In general, we could add more parameters to either define global transformations, such as affine and perspective transformations, or local transformations, such as distortion. The choice of transformations depends on the need of complexity of the generated patterns. We will include more transforms that CPPN is able to achieve in the appendix.

5.2 Optimizing meta-atoms with CC

The objective of our task is to design metasurfaces whose properties are collectively determined by several meta-atoms. As introduced in the last chapter, using the same ES, we can concatenate the latent vectors to identify all the meta-atoms at once. However, this approach turns to be inefficient when the number of nanostructures increases, or equivalently, the dimension of concatenated vectors grows. Besides, in the design of gradient metasurfaces, where the desired properties of a meta-atoms are only associated with its neighbor, the objective cannot be formulated into a single function. This impedes us from using traditional evolutionary algorithms. Hence, we incorporate the CC to circumvent this problem.

CC is an evolutionary computation framework that divides a large problem into small subtasks and solves them independently [149]. As a nature inspired framework, the algorithm constructs species, a group of population (which is also a subcomponent of whole population), and the species of different kinds evolve independently without interaction with species other than itself. Empirically, despite CC is a greedy algorithm, it can lead to satisfactory convergence in the optimization of large-scale high-dimensional functions where other optimization techniques fail.

In order to identify a meta-molecule with multiple meta-atoms while avoiding exponentially increased simulation time for the entire molecule, we divide the design task of the whole meta-molecule into the design of independent meta-atoms, and iteratively optimize the latent vector of each meta-atom. As shown in the flowchart in Figure 5.3, the algorithm treats the population of latent vectors v of meta-atoms as a species. In each iteration of the evolution, the CC algorithm picks one species for update while assuming all other species are optimized. The algorithm then decodes the latent vectors in the species through the CPPN, evaluates the fitness scores of the decoded meta-atoms, and optimizes the species by performing loops of bio-inspired operations including selection, reproduction, and mu-

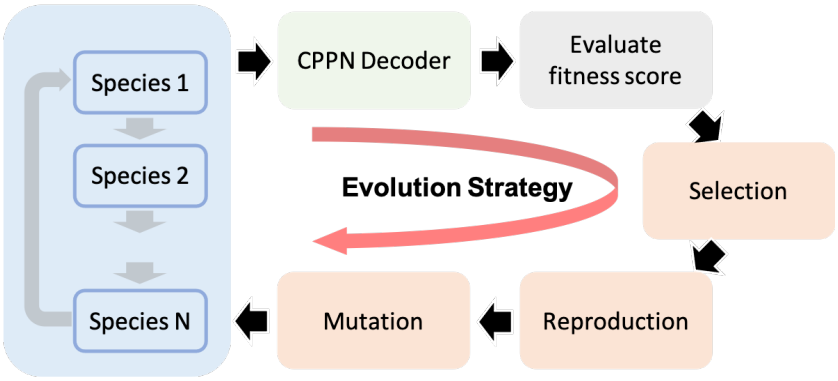


Figure 5.3: The flowchart of the CC algorithm. In each iteration, a species is selected and decoded into meta-atoms. The algorithm then evaluates the scores of all individuals in this species, and performs loops of bio-inspired operations such as selection, reproduction, and mutation. The algorithm iterates in a round-robin fashion until the fitness scores of all species meet the preset criteria.

tation. These operation steps are also adopted in conventional evolution strategies. The CC algorithm performs the above steps and iterates in a round-robin fashion for all species (latent vectors of meta-atoms) until all their fitness scores reach the desired criteria.

5.3 Configuration of meta-atoms and meta-molecules

Our objective is to demonstrate the efficacy of our proposed algorithm for the inverse design of metasurfaces made up of meta-molecules for the manipulation of light. A general lattice schematic of such a metasurface is illustrated in Figure 5.4(a), where the meta-molecule, encircled by dashed lines, consists of multiple distinct meta-atoms indicated by blue, yellow, and orange circles. Figure 5.4(b) illustrates an example of the physical structure of a meta-molecule (diatomic meta-molecule) used in our design. The meta-molecule contains several gold meta-atoms patterned on a glass substrate. Each meta-atom has a thickness of $t = 40\,\mathrm{nm}$ and is located in a square cell with a side length of $p = 320\,\mathrm{nm}$. The periodicity of the meta-molecule in the metasurface is $\Lambda = Np$, where N is the total number of different meta-atoms. In order to accelerate the speed of our framework, we have trained a neural network simulator with $8,000$ full wave simulations of meta-atoms of various shapes. The trained simulator is able to approximate the real and imaginary parts of

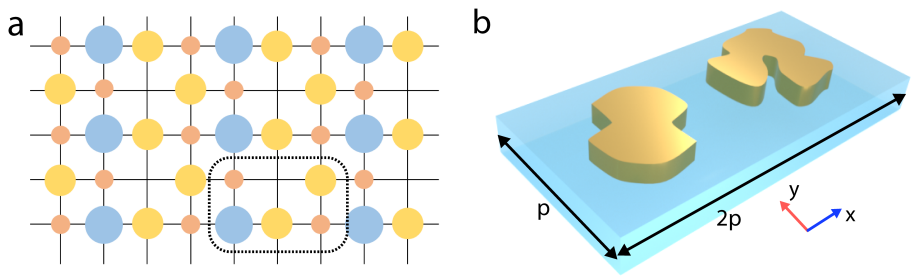


Figure 5.4: Illustration of a meta-molecule in the design problem. (a) The schematic of a two-dimensional lattice of a metasurface, which consists of arrays of meta-molecules. Each meta-molecule, as encircled by the dashed line, contains several distinct meta-atoms. (b) An example of a diatomic meta-molecule comprised of two meta-atoms. In the subsequent studies, the meta-atoms are gold particles with various topologies situated on glass substrate.

the complex transmission and reflection coefficients over a prescribed spectral range, from $170\,\mathrm{THz}$ to $600\,\mathrm{THz}$ as used in the current study, with an accuracy above 97%.

5.4 Design of diatonic meta-molecules for polarization manipulation

We first leverage our framework to design metasurfaces composed of "diatomic" nanostructures for polarization conversion. A diatomic meta-molecule by our definition is a meta-molecule consisting of two distinct meta-atoms. In such meta-molecules, the coupling between adjacent meta-atoms is sufficiently weak, in which case we can approximate the scattered far-field light from the meta-molecule as the superposition of the waves separately scattered by the two atoms. Based on this assumption, we inversely design a series of diatomic meta-molecules that are able to convert linearly polarized incident light into prescribed polarization states at the transmitted side.

5.4.1 Fitness function

We first derive the fitness function in the CC before carrying out the optimization. We denote the set for the index of the meta-atoms as $\vartheta = 0, 1, \ldots, N$, where N is the number of meta-atoms in the metasurface. For $i \in \vartheta$, the fitness functions F_i describe in the following are for the i^{th} meta-atom in the out of the N nanostructures. For the design of

diatomic metasurfaces, we can compute the overall transmittance of the metasurface T as the element-wise linear combination of the transmittances of all meta-atoms:

$$T = \frac{1}{N} \sum_{i \in \vartheta} T_i \tag{5.1}$$

We note that T is a function of wavelength λ and consists of four components – T_{xx}, T_{yy}, T_{xy}, and T_{yx}. The matrix representation is:

$$T = \begin{bmatrix} T_{xx} & T_{xy} \\ T_{yx} & T_{yy} \end{bmatrix} \tag{5.2}$$

The amplitudes and phase of electric fields of the scattered light thus can be derived as:

$$E = \begin{bmatrix} E_x \\ E_y \end{bmatrix} = T \begin{bmatrix} E_x^{inc} \\ E_y^{inc} \end{bmatrix} \tag{5.3}$$

where E^{inc} is the polarization state of the incident light. After obtaining the relative phase difference δ between E_x and E_y, we can derive the polarization angle Ψ and ellipticity χ through the equations:

$$\tan 2\Psi = \frac{2|E_x||E_y|}{|E_x|^2 - |E_y|^2} \cos \delta \tag{5.4}$$

$$\sin 2\chi = \frac{2|E_x||E_y|}{|E_x|^2 + |E_y|^2} \sin \delta \tag{5.5}$$

where $0 \leq \Psi \leq \pi$ and $-\pi/4 \leq \chi \leq \pi/4$. Ellipticity $\chi = 0$ indicates a linear polarized light and $\chi = \pm\pi/4$ represents circularly polarized lights. Suppose the desired polarization states are parametrized as Ψ_0 and χ_0, the fitness function F_i for the ithe meta-atom can be written as:

$$F_i(\Psi, \chi, E) = -\|\Psi - \Psi_0\| - \beta_1 \|\chi - \chi_0\| + \beta_2 \|E\| \tag{5.6}$$

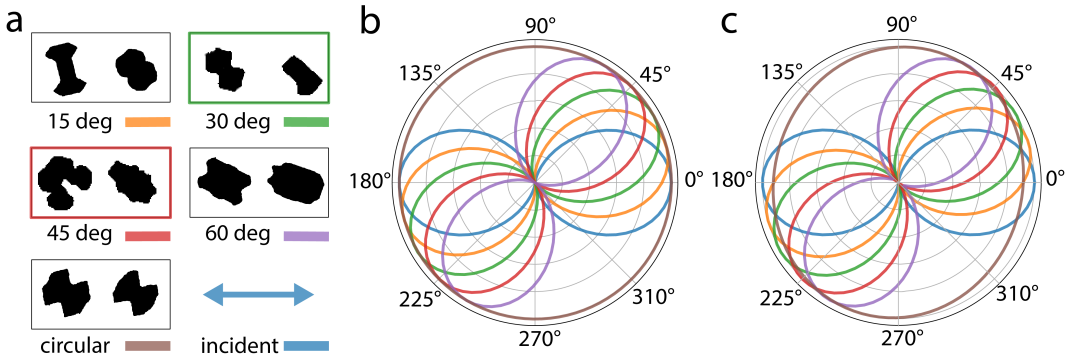


Figure 5.5: Inversely designed diatomic meta-molecules for polarization manipulation. (a) Unit cells of designed diatomic meta-molecules. The design objectives are prescribed polarization rotation/conversion for the transmitted light as indicated underneath each meta-molecule. (b – c) Simulated polarization states of transmitted light from the meta-molecules in (a) with a neural network simulator and FEM full-wave simulation, respectively.

where β_1 and β_2 are penalties. The last term drives the algorithm to identify metasurfaces with large transmittance. The fitness functions for the reflectance follows the same manner.

5.4.2 Design results

Figure 5.5(a) presents the designed meta-molecules that rotate the polarization angle of x-polarized incident light to $15°$, $30°$, $45°$, and $60°$, as well as the design that converts the incident light to a circularly polarized (CP) light, operated at a wavelength of $800\,\mathrm{nm}$. Each design can be accomplished using our framework within 20 seconds on a machine with a single Quadro P5000 GPU. The plots of the converted polarization, computed by the network simulator and FEM full-wave simulation, are illustrated in Figure 5.5(b) and (c) respectively. The intensities of the light are normalized to the maximum of each case to clearly display the polarization states; the plots of actual intensities are also provided in Figure 5.6. As our designed metasurfaces are ultrathin metallic metasurfaces composed of discrete meta-atoms, they inevitably suffer from compromised performance. Circumventing this problem requires the metasurfaces to be multi-layered or dielectric. Nevertheless, our framework is able to identify the structures that can accurately rotate or convert the polarization state, as demanded by the design objectives, in terms of both rotation angle

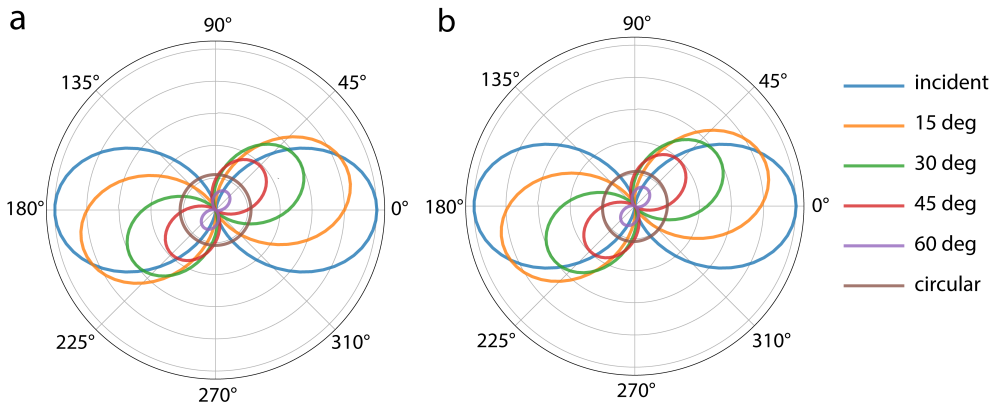


Figure 5.6: Polarizations with actual intensity in correspondence to Figure 5.5. The calculated polarizations of the incident light (blue) and transmitted lights from the network simulator and finite element method simulation, respectively.

and ellipticity.

5.4.3 Experimental verification

We performed nanofabrication and optical characterizations of selected meta-molecules, as inversely designed using the hybrid artificial intelligence framework, to further validate the strategy we have developed. Specifically, here we present case studies for diatomic metasurfaces that can convert x-polarized ($0°$) incident light into linear polarization (LP) light with polarization rotation angles of $\Psi = 30°$ and $45°$. The SEM images of the two fabricated samples are shown in Figure 5.7(a) and (d), while the measured polarization states (dashed lines) as compared to the simulated design objectives (solid lines) are provided in Figure 5.7(b) and (e). The experimental results perfectly match the design objectives despite marginal disagreements incurred by error inherent to the fabrications process. To quantify the overall performance of the two diatomic meta-molecules, we simulate the transmitted polarization states denoted by the tuples (Ψ, χ), representing orientation and ellipticity of the polarization, and plot them on a Poincare sphere as illustrated in Figure 5.7(c) and (h). The blue crosses represent the input polarization (horizontal), and the green/red ones denote simulated output polarization states after the metasurfaces. The designed metasurfaces with diatomic meta-molecules transform the horizontal polarization

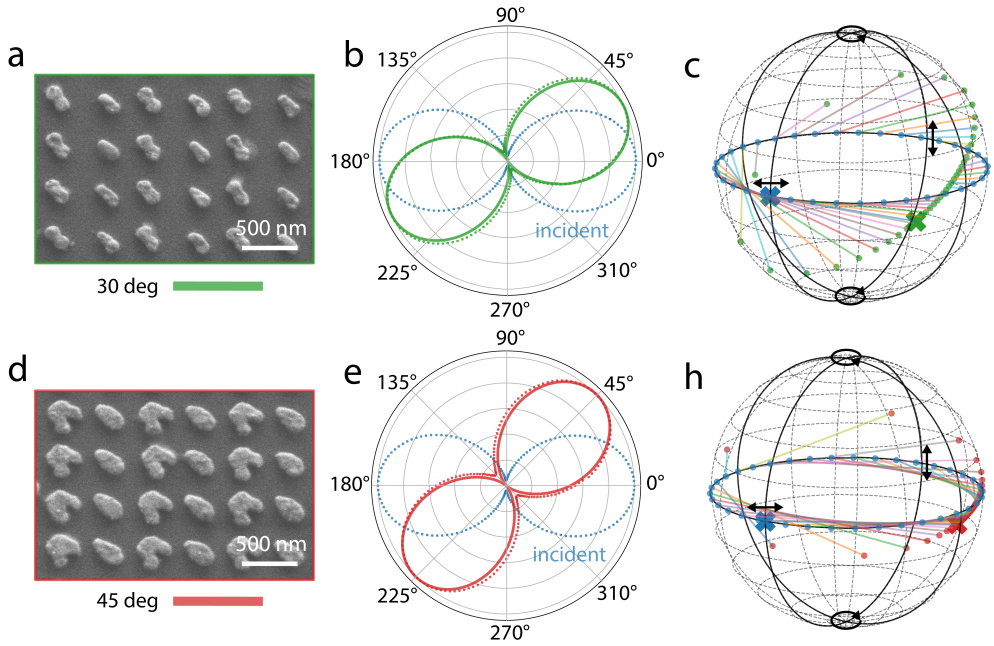


Figure 5.7: (a) and (d) SEM images of the fabricated metasurface polarization rotators with prescribed rotation angles of 30° and 45°, respectively. (b) and (e) Measured polarization states (dashed green/red lines) at the output of the metasurfaces, along with the state of the incident polarization (blue dashed lines at 0°). The solids lines represent the design objectives. (f) and (i) Incident and converted polarization states represented on Poincare spheres. The blue cross indicates the incident horizontal polarization, while the green and red crosses represent the states after the polarization conversion. The transmitted polarization states (green and red dots) at various incident polarization angles (blue dots) are also plotted on the spheres.

at the input to the output states with orientation angles of $\Psi = 31.3^\circ$ and 43.7°, respectively, along with minimal ellipticities, which remarkably match the design objectives with marginal discrepancy. When the incident polarization direction Ψ varies from 0° to 180°, which corresponds to the blue dots on the equator of the Poincare sphere, the polarization states are mapped to a great circle (green and red dots) slighted tilted from the equator.

To clearly illustrate the polarization variation with respect to the incident polarization, we plot the (Ψ, χ) of the output light with respect to the incident polarization orientation as shown in Figure 5.8(a) and (b). Interestingly, the two polarizers have the ability transform the incident polarization angle to the angles towards the design objective (Ψ_{obj}, χ_{obj}). The shadow areas represent the angle regions where the output polarization lies in the range

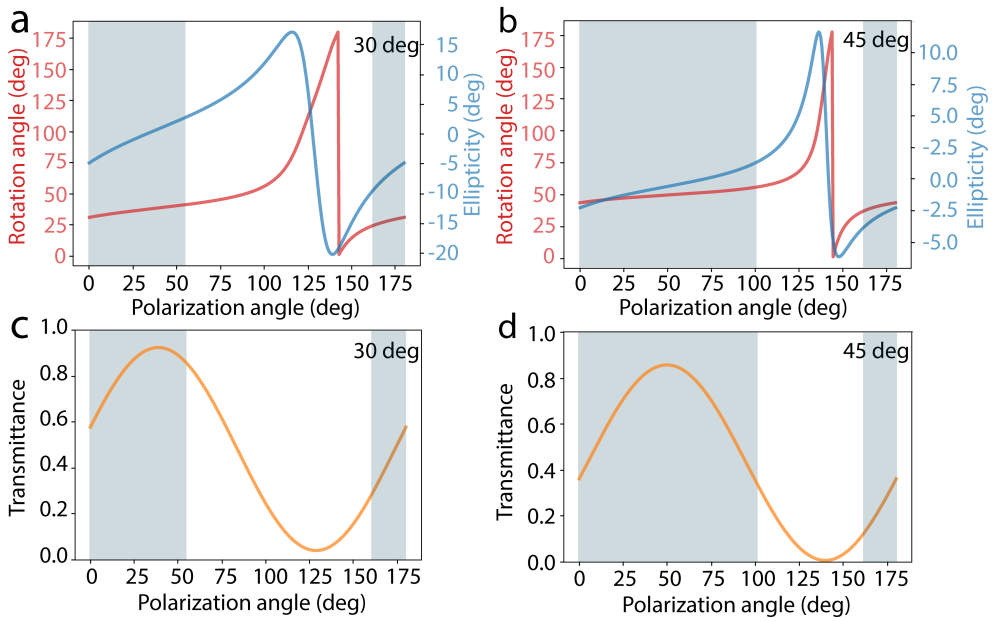


Figure 5.8: Transformed polarizations under various incident polarization angles for the $30°$ and $45°$ diatomic meta-molecules. (a) and (b) The calculated Ψ and χ of the transmitted polarization when incidence is linearly polarized light with orientations from $0°$ to $180°$. (c) and (d) The corresponding transmitted power with respect to the incident polarization angle.

$[\Psi_{obj} - 5°, \Psi_{obj} + 10°]$ with a decent degree of linear polarization (DOLP). In Figure 5.8(c) and (d), we show the transmittance of the light with respect to the incident polarization angle. The overall function of the designed diatomic meta-molecule is to transform the incident polarization to the designed polarization states; if the incident polarization angle exceeds certain threshold, the designed diatomic meta-molecule reduces and eliminates the output power.

5.5 Design of gradient metasurface for propagation manipulation

We further utilize our framework to inversely design metasurfaces with a gradient phase distribution. Such metasurfaces are conventional examples of metasurfaces for diverse functionalities such as the generalized Snell's law, beam steering, and meta-holography. Yet, no existing tools based on machine learning have demonstrated their capacity to tackle this classical problem that requires the design of spatially variant, two-dimensional, ar-

bitrarily shaped meta-atoms and their assemblies. The meta-molecules in our gradient metasurfaces for wavefront control and polarization manipulation are composed of eight meta-atoms.

5.5.1 Fitness function

In a meta-molecule of a gradient metasurface, every two adjacent atoms should scatter light with an equal amplitude and a constant phase difference $\Delta\phi$. This requirement cannot be formulated into a single objective function to be minimized by traditional optimization techniques. To circumvent this problem, we design a fitness function of a meta-atom only associated with itself and its adjacent neighbor. For example, to optimize the i^{th} meta-atom s_i in the meta-molecule, we define its fitness function $F_i(s_i, s_{i-1})$ which is only locally associated with the $(i-1)^{th}$ meta-atom. As a greedy algorithm, the CC does not guarantee optimal solutions, but empirically, it converges to a solution with the desired amplitude requirement and phase distribution in our experiments.

To derive the fitness function for the design of gradient metasurfaces, we first need to compute transmittance T_i of each meta-atom separately. The scattered electric field from the i^{th} element can be obtained as $E_i = T_i E^{inc}$. Without the loss of generality, we assume only y-polarized scattered light $E_{i,y}$ are considered with phase gradient. Denoting the absolute phase of $E_{i,y}$ as $\phi_{i,y}$, then the fitness function for the i^{th} meta-atom can be written as:

$$F_i(E_{i,y}, \phi_{i,y}) = -\|E_{i,y} - E_{i-1,y}\| - \beta_1\|\phi_{i,y} - \phi_{i-1,y} - \Delta\phi_i\| + \beta_2\|E_{i,y}\| \tag{5.7}$$

where $\Delta\phi_i$ is the required phase difference between the i^{th} and $i-1^{th}$ meta-atoms. For metasurfaces with uniform phase gradient, $\Delta\phi_i = 2\pi/N$.

When designing gradient metasurfaces for CP light, after retrieving T_i, we should transform it to the coordinate system where circular polarizations are basis vectors. The trans-

form matrix takes the form:

$$M = M_0 \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix} \tag{5.8}$$

where M_0 is a normalization constant, and its value depends on the definition of the basis vector of CP light. The transmittance in the CP-basis coordinate amounts to:

$$T_i^{cir} = MT_iM^{-1} \tag{5.9}$$

The fitness function for CP gradient-metasurface take the similar form and the rest of the algorithm shall be the same as the one for the identification of linearly polarized gradient-metasurface.

In addition to the fitness function formulated above, here we also present an equivalent approach with Fourier transform to construct the fitness function. We denote $k = 2\pi n/\lambda$ as the wave vector, where n is the refractive index of the optical media. The Fourier transform of the output light field E from the metasurfaces can be written as:

$$\hat{E}(K) = \int_{-\infty}^{\infty} E(x) \exp(-ikx) dx \tag{5.10}$$

where x is the spatial coordinate. This equation can be used for general inverse design of gradient metasurfaces with arbitrary phase profile. Since we are designing a periodic metasurface, we can instead calculate the light intensity of different diffraction orders using Fourier coefficient. We denote the lattice period of the metasurface as Λ, grating vector as $K = 2\pi/\Lambda$, then the coefficients c_n for $n \in \mathbb{Z}$ amount to:

$$c_n = \frac{1}{\Lambda} \int_{-\Lambda/2}^{\Lambda/2} E(x) \exp(-inKx) dx \tag{5.11}$$

In the equation above, c_n can be interpreted as the amplitude of the nth order diffractive

light. We further discretize the equation and approximate the Fourier coefficients as:

$$c_n \approx \frac{\Delta x}{\Lambda} \sum_{i \in \vartheta} E(x_i) \exp(-inKx_i) \tag{5.12}$$

where x_i is the position of the i^{th} meta-atom, and $\Delta x = \Lambda/N$ is the side length of the square where meta-atom situates. In general, to design the metasurface with arbitrary diffractive efficiency for every diffraction order, the fitness function of the i^{th} meta-atom in a metasurfaces can be written as:

$$F_i(c_0, c_1, c_{-1}, \dots) = \sum_{n \in \mathbb{Z}} \beta_n \|c_n\| \tag{5.13}$$

The form of the fitness functions is same for all the meta-atoms. In specific, in order to identify the metasurfaces with maximum first-order diffraction while keep other orders minimum, the fitness function for the ith meta-atom can be written as:

$$F_i(c_0, c_1, c_{-1}, \dots) = \beta_1 \|c_1\| - \sum_{n \in \mathbb{Z} \setminus \{1\}} \beta_n \|c_n\| \tag{5.14}$$

For the metal gradient metasurfaces, it is recommended to set the penalty β_0 small since the zeroth order light is unavoidable.

5.5.2 Design results and experimental verification

We apply our framework to the design of an eight-atom meta-molecule, which can convert x-polarized incident light to y-polarized counterpart, and deflect the cross-polarized light to a specific angle. We choose the central operating wavelength of the metasurface as $800\,\mathrm{nm}$ to facilitate further experimental verifications. Given such requirements, our framework is able to identify a set of meta-atoms satisfying the design objective within 10 minutes on a GPU machine. Figure 5.9(a) shows the unit cell of an identified metasurface, together with the SEM image of the corresponding fabricated sample. We conducted polarization

analysis of the diffracted light on the transmitted side, as illustrated in Figure 5.9(b), where the red dashed line indicates the measured polarization state. As compared to the incident polarization (blue dashed line), the converted polarization is perfectly orthogonal to the incident one as demanded by the design objective. The measured transmittance of the deflected light is 10.9%. A detailed comparison of the simulated and measured polarizations of transmitted light are displayed in the Supporting Information. Figure 5.9(c) presents the FEM simulated distribution of the electric field E_y emerging from the metasurface, which further illustrates the accurate phase gradient and amplitude distribution induced by the eight meta-atoms.

Another example we present here is the inverse design and experimental verification of a gradient metasurface for CP light. The envisioned metasurface should be able to convert LCP light to its cross-polarization (RCP) and bring about an additional phase gradient to the converted portion of light for beam steering. At the design wavelength of $800\,\mathrm{nm}$, one of the identified solutions and its fabricated sample are shown in Figure 5.9(d). We characterize the polarization states of the incident (blue) and converted (red) light in the polar graph as in Figure 5.9(e), where the dash-dotted and dashed lines indicate the measured intensity with and without a quarter waveplate, respectively. The measured transmittance of the RCP light is 9.89%. The orthogonality of the polarizations measured with a quarter waveplate unambiguously proves the sample flips the circular polarization from LCP to RCP. We also provide the simulated converted electric field E_{RCP} under the LCP incidence, confirming the correct polarization, phase gradient, and amplitude distribution of the design. Unlike traditional circular gradient metasurfaces, where only geometric phase contributes to the phase distribution, the phase gradient of our design is jointly induced by the geometric phase ϕ_{geo} and the material-induced phase delay ϕ_{mat}. When the incidence is LCP, ϕ_{geo} and ϕ_{mat} constructively contribute to the desired phase gradient; however, under the RCP incidence, ϕ_{geo} and ϕ_{mat} destructively contribute to the phase profile, causing asymmetrical behavior of the metasurface. To illustrate this fact, in Figure 5.9(g) we provide the

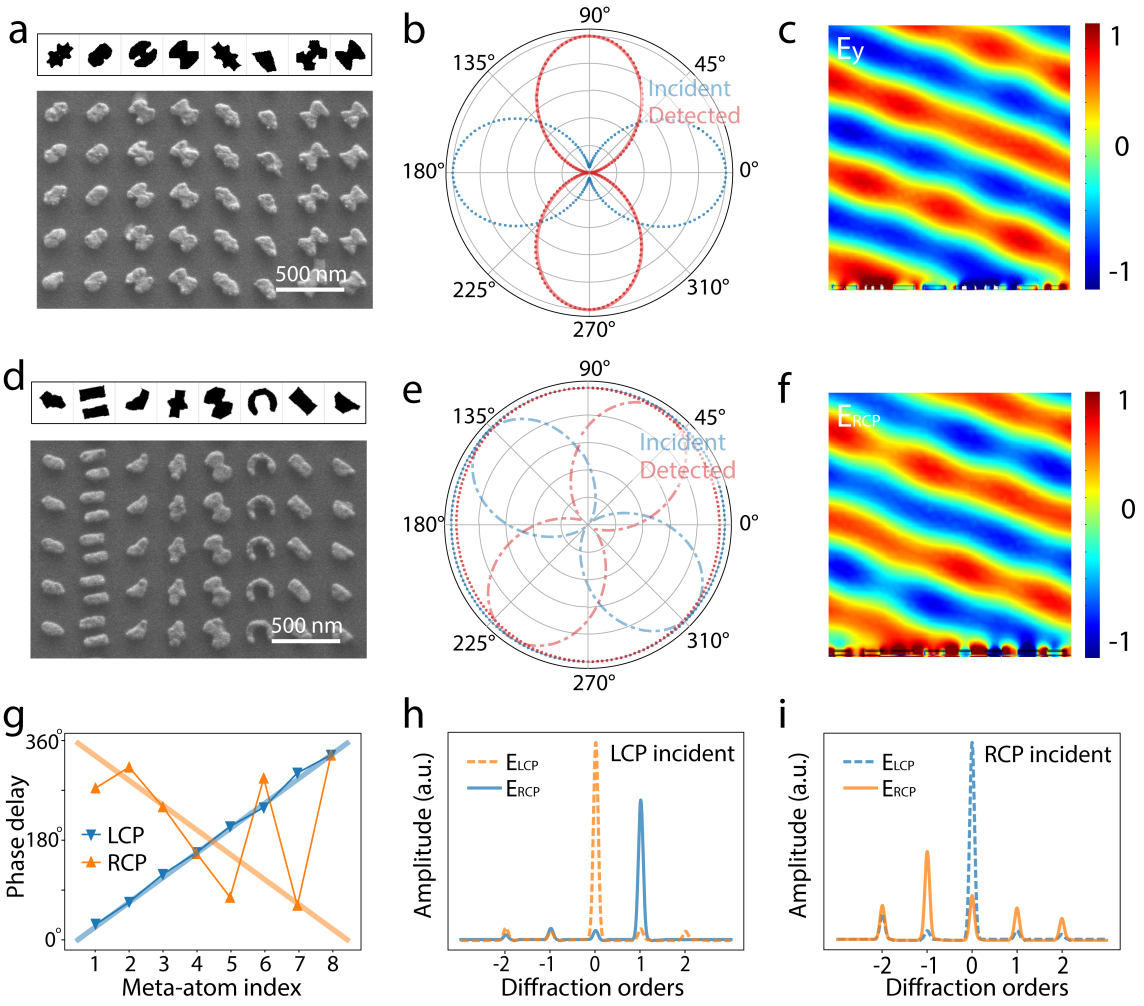


Figure 5.9: Inversely designed meta-molecules for gradient metasurfaces. (a) Unit cells and SEM image of a gradient metasurface for polarization conversion and beam steering with linearly polarized incidence. (b) Measured polarization states of the incidence (blue dashed line) and the diffracted beam (red dashed line). The solid red line (which essentially overlaps with the experimental data) indicates the design objective. (c) Simulated electric field E_y distribution in the proximity of the metasurface shown in (a) under x-polarized incident light. (d) Designed gradient metasurface for polarization conversion and beam steering with LCP incidence. (e) Measured polarization states (dashed lines) of the incident and diffracted lights. The dash-dotted lines represent measured data after a quarter waveplate. (f) Simulated electric field E_{RCP} distribution under the LCP incidence. (g) Relative phase delay induced by the eight meta-atoms shown in (d) for LCP and RCP incident waves, respectively. The solid lines represent the phase distribution of conventional geometric-phase-based circular metasurfaces. (h – i) Simulated diffraction behavior of the metasurface under LCP and RCP incident lights, respectively, where circularly polarizations of opposite handedness are steered to different directions. Note that the LCP incidence is the designed operating mode for this metasurface.

phase delay induced by the eight meta-atoms under circularly polarized incidence. Unlike the case of LCP incidence (blue markers, which is the polarization this metasurface is designed to serve), where the phase delay parallels the linear distribution of a conventional circular gradient metasurface, the phase shift with the RCP incidence (orange markers) deviates from the constant phase gradient, causing multiple diffractions. This property is also reflected in the calculated far-field angular distributions presented in Figure 5.9(h) and (i). Since our framework does not consider the particular physical mechanism during the inverse design process, nor does it require predefined constraints or human intervention, it tends to discover metaphotonic devices and novel photonic phenomena with complex light-matter interactions.

5.6 Discussion and conclusion

In summation, we have proposed a framework consolidating a CPPN generator and a cooperative coevolution algorithm to resolve the inverse design of meta-molecules comprised of spatially variant meta-atoms. The CPPN generator creates high-quality nanostructure patterns with complicated features, while the CC algorithm expedites the identification of the multi-element metasurfaces by dividing the overall design problem into individual subtasks. With the acceleration of a neural network simulator, our framework is able to discover novel metasurfaces with prescribed design objectives, as exemplified in the inverse design and experimental verification of gradient metasurfaces for polarization and wavefront control. Since the proposed framework relies on stochastic evolutionary algorithms, it tends to converge to a solution near the global minimum under several, if not one, fast runs with random initializations. The cost of failure design is also dramatically reduced due to the inexpensive simulation enabled by the neural network simulator. By the nature of CPPN, the topologies of the designed patterns are fully dependent on the user-defined geometric dataset. By including simple geometries in the dataset, our framework is able to identify nanostructures similar to the specified topologies. This feature can be utilized for

constraining the geometric complexity of nanostructures for fabrication purposes. On the other hand, relying on the short runtime and stochastic property of the framework, easily fabricable and manufacturable samples can be selected from multiple optimized structures after several fast runs of the algorithm. As demonstrated in our paper, a well-trained neural network simulator is reliable enough for the practical application of the framework, but we would encourage to replace the simulator with a physical simulation approach at the last few optimization iterations. This will eliminate occasional flawed solutions incurred by the inaccurate approximation of the neural network simulator. In addition to the case studies we have demonstrated here, the proposed strategy can be applied to the design of a wide range of metasurfaces and photonic devices whose overall performance can be predicted by the properties of their subcomponents such as multi-layered metasurfaces and achromatic metalenses. Thanks to the fast inference speed of neural networks, our algorithm is also capable of exploring unconventional optical phenomena induced by proper combination of multiple meta-structures with little cost of failed discoveries. In terms of the framework itself, as the ES optimization does not require gradients from the fitness function, the formulated design objectives can be arbitrary with little constraints. This unique feature enables the framework to be a powerful tool for general inverse design tasks aiming at on-demand, user-defined design objectives in terms of scattering, dispersion, chirality, absorption, nonlinear responses, and nearfield enhancement. Furthermore, since the proposed framework is an implicit method without consideration of the actual dynamics of the systems, the strategy developed here presents a broad and versatile potential for the exploration of novel phenomena and prototypes in generic design problems in other disciplines, such as materials science, condensed matter physics, and chemical syntheses.

CHAPTER 6
TOPOLOGICAL ENCODING METHOD FOR DATA-DRIVEN PHOTONIC DESIGN

Optimizing the topology of a photonic structure with arbitrary shape has been a long-sought-after goal. Typically, the topology of photonic structures is represented in binary images. Because of the discretization and the high DOF of binary images, optimization is likely stuck in local minima. Although generative models are able to discover new topologies so as to approach the global minimum, the bias of the training dataset and the limited capacity of the network cause incompleteness of the solution space, i.e., the global minimum may not be included in the space defined by the training dataset. Here, we propose an encoding method that is able to transform the binary image to a continuous sparse representation with maximum physical information maintained. This encoding approach can be used for data generation and dimensionality reduction of photonic structures, augmenting the capacity of machine learning models for analyzing the dataset, and enhancing the likelihood of achieving the global minimum in the optimization problems.

6.1 Encoding the topology of photonic structures with Fourier transform

Our goal is to encode the nanostructures represented in a 2D binary image into a continuous sparse representation so as to assist data-driven approaches such as macnine learning and deep learning to analyze, discover, and optimize the topology of photonic structures. In the following discussion, we denote the binary image to be encoded as $f(x, y) \in \{0, 1\}^N$, where x and y are coordinates of the image, and N is the dimension of the image. Fourier transform (FT) can be used to transform the binary image to a sparse representation. However, when some operations such as filtering out high-frequency components, are applied to the transformed sparse representation, the inverse Fourier transform (IFT) of the sparse

representation may not be a binary image anymore. Although a threshold can be used to binarize the image, different thresholds can transform a non-binary image to different binary images. An inappropriate choice of the threshold may result in the incorrect retrieval of the desired topology. Consequently, FT is not able to encode, decode, and manipulate arbitrary topologies of photonic structures for the general purpose of dimensionality reduction, data analysis, and device optimization.

Instead of simply applying FT to the binary image, we carry out FT to the level set function $\phi(x, y)$ that defines the topology of the structure. A level set function $\phi(x, y)$ is defined as a 3D surface parametrized by the coordinates x and y. The topology of a photonic structure can be represented by the zero-level set of $\phi(x, y)$ as $\Gamma = \{(x, y)|\phi_e(x, y) = 0\}$. Given a certain binary image as shown in Figure 6.1(a), our encoding strategy is first to construct a level set function as illustrated in Figure 6.1(b), and then to apply FT to the level set function so as to derive the sparse representation of the binary image as presented in Figure 6.1(c). Inversing the whole process reconstructs a sparse representation to a binary image. The detailed procedure of encoding and decoding photonic structures is shown in Figure 6.1(d). We first construct the level set function of the original image $f(x, y)$ through the transform:

$$\phi_e(x, y) = e^{i\pi f(x,y)} \tag{6.1}$$

The exponential function maps the original image to the discrete value $\{-1, 1\}^N$, so that the topology of the structure can be represented by $\Gamma = \{(x, y)|\phi_e(x, y) = 0\}$. By carrying out FT to ϕ_e, we can find the sparse representation of ϕ_e in the complex frequency space as

$$\hat{\phi}_e(k_x, k_y) = \mathscr{F}[\phi_e(x, y)] \tag{6.2}$$

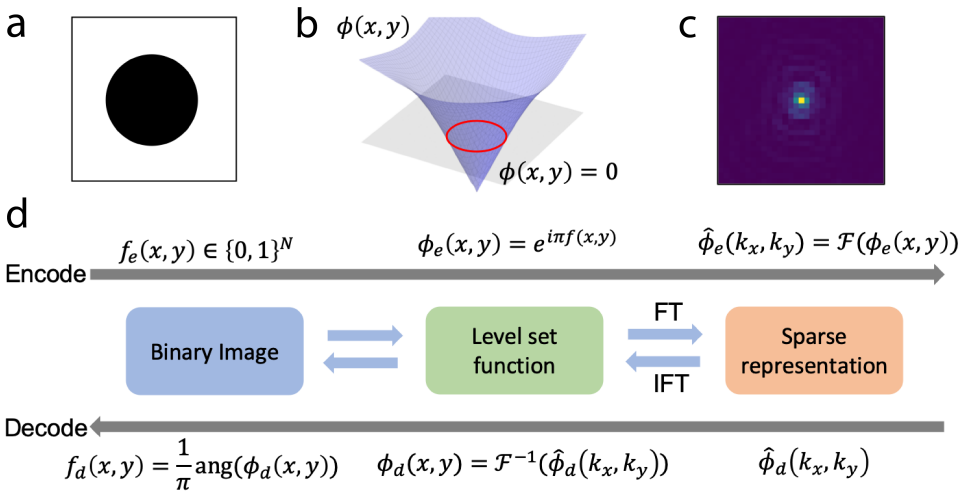


Figure 6.1: Description of the encoding method (a) Binary image representation of a nanostructure. (b) Illustration of a level set function $\phi(x, y)$. The topology shown in (a) (encircled by red line) is represented by the zero-level set. (c) Encoded sparse representation of the nanostructure shown in (a). (d) The outline of the encoding method. For the encoding process, a level set function is first constructed from the given binary image. The spare representation of the image is derived from the Fourier transform of the level set function. Decoding the binary image from the sparse representation is the inverse of the encoding process.

By the property of FT, $\hat{\phi}_e$ naturally satisfies the condition:

$$\hat{\phi}_e(k_x, k_y) = \hat{\phi}_e^{\dagger}(-k_x, -k_y) \tag{6.3}$$

and then retrieve the binary image representation through the operation:

$$f_d(x, y) = \frac{1}{\pi}\,\text{ang}(\phi_d(x, y)) \tag{6.4}$$

where $\text{ang}(\cdot)$ is a function that calculates the phase of $\phi_d(x, y)$, and $1/\pi$ is the normalizer that ensures the retrieved image has binary values 0 and 1. Note Eq. (4) is identical to setting a threshold $\phi_0 = 0$ and retrieving the image by

$$f_d(x, y) = \begin{cases} 1 & \text{if } \phi_d(x, y) > \phi_0 \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

6.2 Properties of the encoding method

The encoding method illustrated above transforms a binary image to a continuous sparse representation, allowing incremental variation of the topology of the structures by perturbing $\hat{\phi}_d(k_x, k_y)$. We will illustrate several properties of this encoding method and show its advantages in representing photonic structures for data-driven photonic discovery. To be consistent with terminology in machine learning, we will call the space of the sparse representation a latent space and $\hat{\phi}_d$ will be a latent vector.

6.2.1 Dimensionality reduction

Suppose a photonic structure is represented in an image with N number of pixels, and also the DOF of N. Performing inverse design and optimization in such a high-dimensional space is difficult. On the other hand, the topology of a structure usually presents some properties such as continuity and connectiveness for the purpose of proper simulation and fabrication. Thus, the available structures cluster in a small region in the N-dimensional image space. Machine learning algorithms such as VAE and GAN have been used to reduce the dimensionality of the photonic structures. However, a trained machine learning model can only faithfully encode and decode the structures topologically similar to the training dataset. Our method, derived without the dependence of data, is able to process binary image data in a fast and general manner.

Figure 6.2(a) to (d) illustrate the process of encoding and decoding a photonic structure. The initial structure, as shown in Figure 6.2(a), is represented in an image with $N = 64 \times 64$. Our method encodes the structure into its frequency representation $\hat{\phi}_e$, the norm of which is presented in Figure 6.2(b). After cropping the high-frequency components as in Figure 6.2(c), only dominant components are kept in $\hat{\phi}_d$. Recovering $\hat{\phi}_d$ into binary images through our method, we achieve the decoded image. Interestingly, as decoding the structure is essentially performing the IFT, the recovered images can have arbitrarily large

DOF. The decoded example shown in Figure 6.2(d) has a resolution of 128×128. As cropping inevitably deletes a portion of the information in the latent space, our encoding method is an irreversible lossy compression. The higher order of terms kept in the latent space, the finer feature will be remained in the decoded images. Other dimensionality reduction approaches can be applied to the encoded latent vectors to further reduce the complexity of the inverse design problem.

6.2.2 Continuity of the latent space

Since FT and IFT are uniformly continuous operators, the decoded image $f_d(x, y)$ can be incrementally varied by perturbing $\hat{\phi}_d(k_x, k_y)$. Figure 6.2(e) to (i) show continuous topological variation from the first pattern shown in Figure 6.2(e) to the last one in Figure 6.2(i). The first and last patterns are randomly constructed with latent vectors $\hat{\phi}_1, \hat{\phi}_2 \in \mathbb{R}^{7\times7}$. Figure 6.2(f) to (h) present the intermediate that are decoded from the linearly interpolated latent vector $\hat{\phi}_d = \lambda\hat{\phi}_1 + (1 - \lambda)\hat{\phi}_2$, where $\lambda \in (0, 1)$. As we can observe, the two distinct topologies can be smoothly transformed by linearly interpolating their latent vectors. This property is indispensable for the fast convergence when ES is utilized for the topology optimization [**my paper**]. It is noteworthy that linear interpolation does not always result in continuous topological transformation, especially when the latent space is high dimensional and the topologies of the two patterns are significantly distinct. In this situation, the geodesic, representing the shortest path between the two patterns in the latent space, should be computed for the smooth transformation.

6.2.3 Symmetry

Symmetry is a crucial geometric property that should always be considered in the design of photonic devices. Properly leveraging the symmetry of the photonic structure reduces the time of simulation and mitigates the difficulty of optimization. Our encoding method maintains the symmetry of binary images in the frequency spaces. Figure 6.2(j) to (n) display

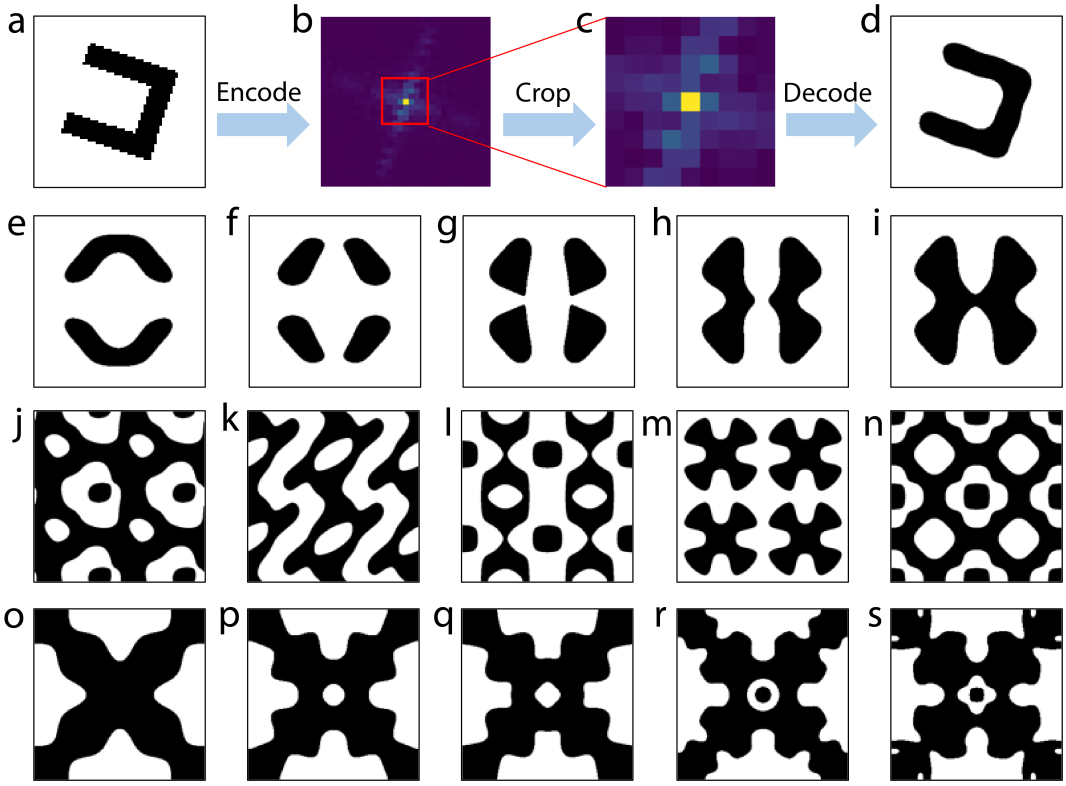


Figure 6.2: Properties of the encoding method (a) – (d) Dimensionality reduction using the proposed encoding method. Initial binary image (a) is encoded to the sparse representation (b). The low-dimensional representation (latent vector) can be achieved by deleting the high-frequency components as shown in (c). The latent vector can be recovered to the initial structure without substantial loss of information. (e) – (i) Continuously varying two topologies by linearly interpolating the latent vectors. (j) – (n) Generated samples with various geometric symmetries. The shown images are tiled unit cells of the generated patterns. (o) – (p) Adding fine features to initial pattern (o) by gradually expanding the dimensions of latent vectors from 7×7 to 15×15.

a few randomly generated patterns with various geometric symmetries. The dimension of the latent space we chose is $N = 5 \times 5$. Without any constraints on the latent space, the generated pattern shown in Figure 6.2(j) does not present any symmetry. In order to generate a centrosymmetric pattern, we need to enforce the latent vector to be centrosymmetric, i.e., $\hat{\phi}_d(k_x, k_y) = \hat{\phi}_d(-k_x, -k_y)$. Combined with Eq. (3), this condition is equivalent to the latent vectors being real, reducing the DOF of the latent vectors to $\lfloor N/2 \rfloor + 1 = 13$. Figure 6.2(k) is a randomly generated centrosymmetric pattern with such constrain. Similarly, if $\hat{\phi}_d(k_x, k_y) = \hat{\phi}_d(k_x, -k_y)$ and $\hat{\phi}_d(k_x, k_y) = \hat{\phi}_d(-k_x, k_y)$ are enforced, we can generate axisymmetric patterns such as the one shown in Figure 6.2(l). The DOF in this case is reduced to $\lfloor \sqrt{N}/2+1 \rfloor^2 = 9$. By additionally constraining $\hat{\phi}_d(k_x, k_y) = \hat{\phi}_d(k_y, k_x)$, axisymmetric patterns with axis of symmetric $y = x$ can be produced as shown in Figure 6.2(m) and (n). In this circumstance, the DOF of the pattern is $(\lfloor \sqrt{N}/2 \rfloor+1)(\lfloor \sqrt{N}/2 \rfloor+2)/2 = 6$, indicating that only six variables are required to arbitrarily manipulate the topology of photonic structures. This unique feature of the encoding/decoding method can be leveraged for reducing the parameters in the inverse design of metasurfaces and photonic crystals with specific polarization requirements.

6.2.4 Multilevel optimization

When the DOF of a photonic structure is large, optimization techniques suffer from problems such as slow convergence and local minimum. In this situation, multilevel optimization[150] can be used for designing the structure and enhancing the performance. By the nature of FT, our encoding method allows the multilevel optimization of photonic structures by gradually modifying the corresponding latent vectors. Figure 6.2(o) to (s) present an example of adding finer features to the initial structure (Figure 6.2(o)) through our encoding method. The initial structure is constructed from a $\hat{\phi}_d^0 \in \mathbb{R}^{7\times7}$. By attaching additional vectors to the latent vector, we can augment the latent vector to $\hat{\phi}_d^1 \in \mathbb{R}^{9\times9}$ in a higher dimension. Figure 6.2(p) presents the decoded image of the augmented latent vector $\hat{\phi}_d^1$.

A few features such as the hole in the center appear. Repeating the augmentation process results in the incremental evolution of the structures with finer features as shown in Figure 6.2(q) to (s). This unique property of the encoding method enables the consolidation of traditional optimization and multilevel optimization for the inverse design of high DOF photonic structures.

6.3 Designing non-paraxial DOE

As a case study, in this section we will represent how the encoding method can be applied in the inverse design and optimization of binary DOEs. Traditionally, the design of DOE have relied on an iterative Fourier transform algorithm (IFTA) [151, 152]. The algorithm can generate binary phase masks whose FT is proportional to the required diffraction intensity distributions. However, IFTA does not take into account the actual physical process, resulting in the inaccuracy of the modeling for diffraction intensity. This inaccuracy prevents an effective design method for the non-paraxial diffractive beam splitters. To solve this problem, we consolidate the proposed encoding method and the hybrid inverse design algorithm introduced in Chapter 4 to design 3×3 non-paraxial diffractive beam splitters with various diffraction intensity distributions.

6.3.1 DOE configuration

The cross section of a common configuration of the DOEs is shown in Figure 6.3(a). The grating and the substrate share the same material with a refractive index $n = 1.56$, and the thickness of the grating pattern is $t = 840\,\mathrm{nm}$. Monochromatic light with a wavelength of $\lambda_0 = 940\,\mathrm{nm}$ is incident from the substrate side. We set the period of the DOE as $p = 2.83\,\mu\mathrm{m}$, so that the angle between the 1st and the 0^{th} order diffraction is about 21°. The objective of the design is to identify the DOE patterns that are able to accurately diffract the incident light into the 9 different directions with various required intensity distributions. We note that under the incident wavelength of $940\,\mathrm{nm}$ there are total 5×5 diffraction

orders generated from the DOE. We aim to optimize the intensity distribution of the central 9 orders, while treating the other orders as noises and suppress the intensities of them.

6.3.2 DOE design strategy

In order to leverage data-driven approaches such as deep learning for fast global optimization, we first generate sufficient DOEs that are able to diffract light to the desired directions. Since the encoding method we proposed is based on FT, it is sufficient to sample $\hat{\phi}_d$ from $[-1, 1]^{3\times3}$ as the sparse representation for the design of 3×3 diffractive beam splitters. In practice, we can further simplify the representation of each DOE for the convenient training of a neural network model. In detail, we write $\hat{\phi}_d$ as:

$$\hat{\phi}_d = \begin{bmatrix} v_1 + iv_2 & v_3 + iv_4 & v_5 + iv_6 \\ v_7 + iv_8 & v_9 & v_7 - iv_8 \\ v_5 - iv_6 & v_3 - iv_4 & v_1 - iv_2 \end{bmatrix} \tag{6.6}$$

where $\{v_i | i = 1 \dots 9\}$ are real numbers. Equation (7) satisfies the condition defined by Equation 6.3 and has a DOF of 9. Sampling a random $\hat{\phi}_d$ is achieved by independently sampling each v_i from a uniform distribution. We reorganize the entries in $\hat{\phi}_d$ to an encoded vector $v = [v_1, v_2, \cdots, v_9]$, and take the encoded vector as the input of the network. For each encoded vector v, we simulate the diffraction efficiencies η of corresponding grating structures with RCWA [153]. Under the wavelength of incident light $\lambda_0 = 940$ nm, the DOEs produce 5×5 orders of diffraction light. Next, the simulated diffraction intensities are reorganized into a 25-dimensional vector, and normalized with respect to the largest intensity η_{max}. The final representation of each simulated result is denoted as a 26-dimensional vector:

$$K = \left[\frac{\eta_{-5,-5}}{\eta_{max}}, \frac{\eta_{-5,-4}}{\eta_{max}}, \cdots, \frac{\eta_{5,5}}{\eta_{max}}, \eta_{max} \right] \tag{6.7}$$

As such, (v, K) is a training pair of the network model. To construct the whole dataset for

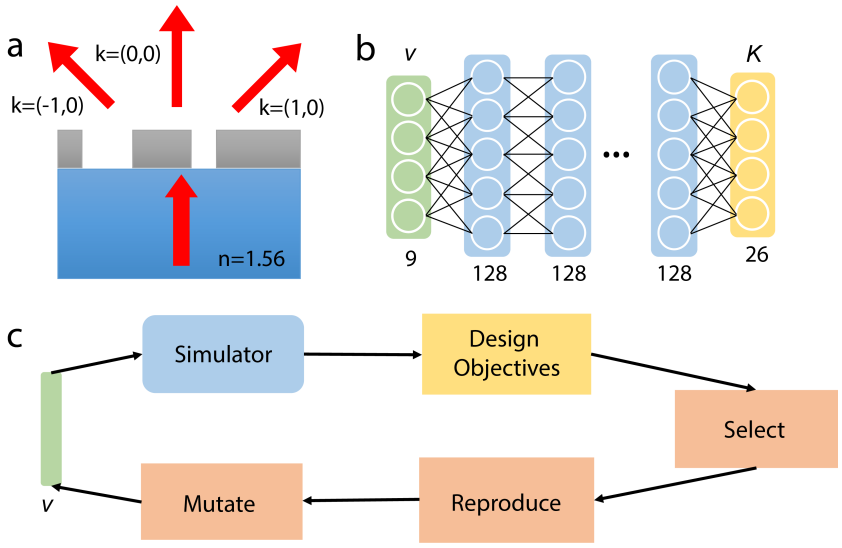


Figure 6.3: Training of the network and statistic of the optimized results (a) The variation of training (blue) and validation (orange) loss versus the training epoch. The validation loss reaches 0.03 after 100 epochs of training. (b) Uniformity errors of 150 designed DOE structures with the objective of all diffraction intensities being equal. The blue bars represent the distribution of U_{err} calculated with the network simulator during the optimization, and the oranges bars are the evaluated U_{err} using RCWA for validation. The final design is selected from the validated DOE with the minimum U_{err}.

the training of the network, we randomly sampled 15,000 encoded vectors v and performed the process outlined above. We split the dataset into two parts with 12,000 for training and the rest for validation. Since the grating patterns are represented in low-dimensional vectors v, a simple neural network architecture is sufficient for accurate approximation of the diffraction efficiency. As shown in Figure 6.3(b), we built an eight-layered fully connected neural network with input of encoded vector v and output of efficiency vector K. All the hidden layers of the network contain 128 neurons, and the nonlinear activations after each input and hidden layers are ReLU. During the training of the simulator, we set the loss function as the MSE of the predicted diffraction intensities $\hat{K}$ with respect to the training data K:

$$L(K, \hat{K}) = \frac{1}{N_K} \sum_{i=1} N_K \|K - \hat{K}\|^2 \tag{6.8}$$

where $N_K = 26$ is the dimension of the vector K. Figure 6.4(a) presents the loss variation during the training process. A validation loss below 0.03 is achieved after 100 epochs of

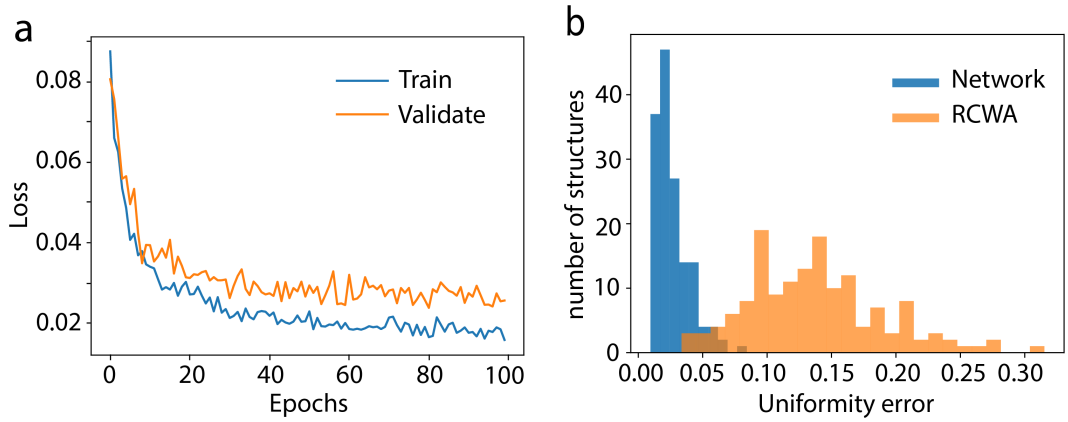


Figure 6.4: Training of the network and statistic of the optimized results (a) The variation of training (blue) and validation (orange) loss versus the training epoch. The validation loss reaches 0.03 after 100 epochs of training. (b) Uniformity errors of 150 designed DOE structures with the objective of all diffraction intensities being equal. The blue bars represent the distribution of U_{err} calculated with the network simulator during the optimization, and the oranges bars are the evaluated U_{err} using RCWA for validation. The final design is selected from the validated DOE with the minimum U_{err}.

training.

In order to globally optimize the topology of the grating patterns, we adopt the modified ES as shown in Figure 6.3(c). The algorithm starts with sampling a population of random encoded vectors v. Each vector is regarded as an individual in the population. These vectors are simulated through the neural network simulator. Based on the simulated results, the population is subsequently evaluated by certain design objectives, and the elites of the population are selected for the following reproduction and mutation. The algorithm iterates until one of the individuals achieves the design criterion or the maximum iteration is reached. To design DOEs with various diffraction intensity distributions $\eta_{obj} \in [0, 1]^{3\times3}$, we define the uniformity error of a design as:

$$U_{err} = \frac{\tilde{\eta}_{max} - \tilde{\eta}_{min}}{\tilde{\eta}_{max} + \tilde{\eta}_{min}} \tag{6.9}$$

where $\tilde{\eta}_{max}$ and $\tilde{\eta}_{min}$ are the maximum and minimum intensity of the scaled diffraction intensity $\eta_{scaled} = \eta/\eta_{obj}$, where η is the simulated performance of the designed beam splitter. Our objective is to minimize U_{err} of a DOE design given certain intensity distri-

bution η_{obj}. As the network simulator is adopted to approximate the U_{err} in the optimization process, the actual intensity distribution of the designed DOEs (which is evaluated by RCWA) may not be the optimal. In order to identify the optimal solution, we carried out 150-time ES-based optimizations with the network simulator, evaluated all the designed structures using physical simulation RCWA, and selected the optimal solution with based on the RCWA-evaluated U_{err}. With the acceleration of the network simulator, each run of the ES-based optimization is within 1 second. Figure 6.4(b) shows the histogram of U_{err} after 150 runs of the optimization when the intensities of all diffraction orders are required to be equal (i.e. all entries in η_{obj} are 1). The blue bars represent the U_{err} of all 150 DOEs calculated through the network during the optimization, and the oranges ones are the U_{err} evaluated by RCWA for the selection of optimal solutions.

6.3.3 DOE design results

Figure 6.5(a) to (h) present eight examples of designed DOE with various intensity distribution η_{obj}. For each panel in Figure 6.5, the left plot is the tiled unit cell of the designed DOE, the middle image shows the simulated diffraction efficiencies of the desired 3×3 orders and other suppressed noises, and the right plot compares the desired diffraction intensities (blue) and RCWA simulated results of the designed DOE (orange). With the help of our encoding method, the hybrid framework successfully identified DOE structures with diffraction intensity distribution matching the design objectives. Quantitively, the uniformity errors U_{err} of designed DOEs from Figure 6.5(a) to (h) are 0.035, 0.045, 0.073, 0.068, 0.194, 0.036, 0.352, and 0.079, respectively. Since U_{err} is calculated through scaled intensity η_{scale} and objective intensity η_{obj} is the denominator of η_{scale}, the error is extremely sensitive to the objective intensities with small values. In the examples shown in Figure 6.5(f), (h) and (i), diffraction intensities in some orders are required to be small. In such cases a tiny disagreement of the actual diffraction and objective intensities induces large U_{err}. Nevertheless, the overall intensity distributions of all designs have excellent agreement

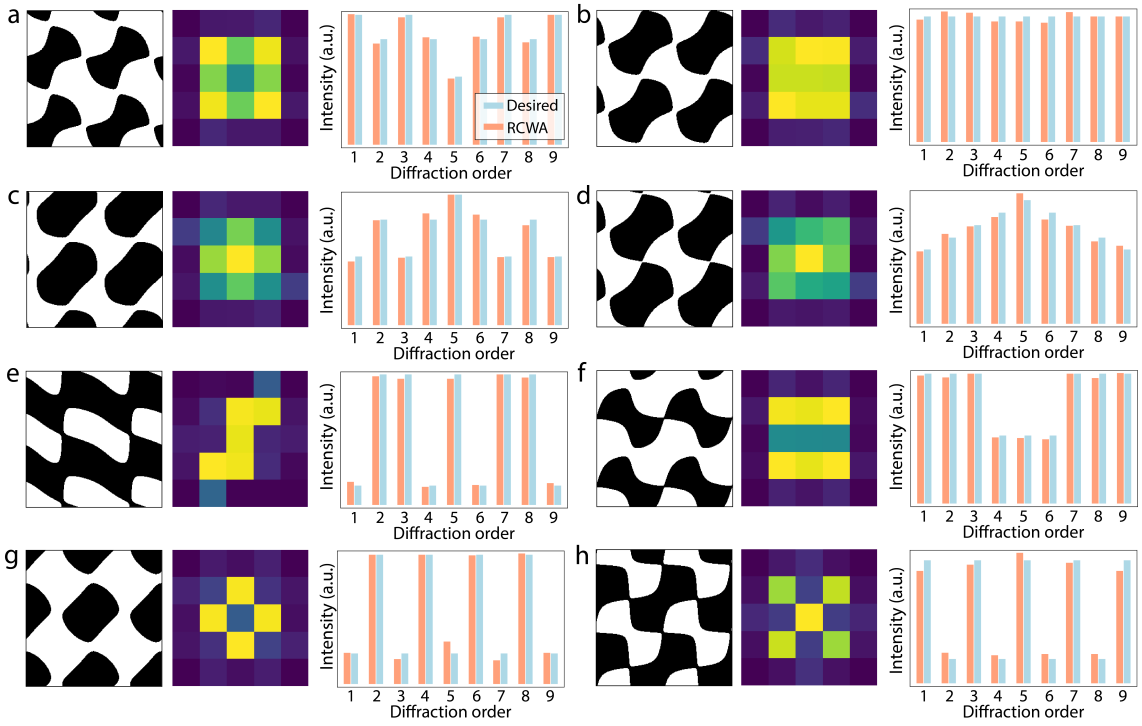


Figure 6.5: Examples of designed DOEs with various diffraction intensity distributions. In each panel, the leftmost figure represents a tiled unit cell of the designed DOE. The middle image represents the simulated efficiencies of all the diffraction orders. The rightmost plot compares the objective intensities (blue) versus the RCWA simulated intensities (orange) of the design. All the designed DOEs are able to diffract light with intensity distributions essentially replicating the design objectives. The uniformity errors of the displayed designs are (a) 0.035, (b) 0.045, (c) 0.073, (d) 0.068, (e) 0.194, (f) 0.036, (g) 0.352, (h) 0.079, respectively. By the definition of Eq. (10), when the design objectives include diffraction orders with small intensities, a tiny disagreement of actual diffraction and objectives induces large uniformity errors. This leads to a large U_{err} for the design shown in (e), (g), and (h).

with respect to the objectives, confirming the effectiveness of our encoding methods in the machine-learning-based inverse design approaches.

6.4 Discussion and conclusion

In summary, we have proposed an encoding method that is able to transform the topology of a photonic structure represented in discrete, high-dimensional, and binary pixelated images into a continuous sparse representation. We explored properties of the encoding methods, such as the continuity of the latent space and geometric properties of the encoded

topologies, and discussed the potential application of this method for dimensionality reduction and data generation for the data-driven photonics optimization. As a case study, we utilized the encoding method and a deep learning-based optimization framework to design 3×3 DOEs with non-paraxial diffraction angle and various diffraction intensity distributions. The encoding method allows us to generate sufficient data for the optimization without exploring unnecessary solution space. The encoded DOE represented in the low dimension also enhances the accuracy of the network and, as a result, the fidelity of the design is increased.

Although the proposed encoding methods are aimed at assisting the data-driven inverse design of photonic structures, other derivative-free traditional optimizations [51] can also take advantage of the continuous low-dimensional representation of photonic structures. If global optimization is not required, local search algorithms can be applied to the latent vectors of the structures without generating a redundant dataset. In the future, we expect to explore the application of the encoding method with both traditional and data-driven optimization approaches for the discovery and design of other photonic media such as photonic crystals and metasurfaces, and anticipate to consolidating the encoding method with deep generative models to produce complex patterns for the general inverse design of photonic structures.

CHAPTER 7
CONCLUSION AND OUTLOOK

In the dissertation, we have presented the development of deep-learning-based strategy for the simulation and inverse design of photonic structures. We first introduced the method of modelling physical systems using deep learning model. In physical research, implementing deep neural networks for the surrogate model needs to take into account of several physical restrictions. Then, we proposed a generative model for the inverse design of photonic structures. The GAN is utilized such that an optimal solution from a geometric dataset can be found in a fast global way. We implement the strategy to design metasurface structures, and the design meta-atoms have marginal different compared to the design objectives. Following the GAN-based approach, we implemented the ES algorithm as the optimizer in the design framework, and successfully demonstrate the efficacy of the strategy through metasurface design for transmittance, polarization, and propagation manipulation. Finally, we proposed an encoding method for data-drive photonic design. Such a method can filter out unnecessary structures in the design of DOEs and increase the likelihood of identifying optimal solutions. At the end of this chapter, we will discuss some challenges, future research, and outlook of the deep-learning-based photonic design.

7.1 Predicting optical responses with discriminative models

As an essential part of the inverse design toolkit, simulation method determines the speed of convergence and thus the likelihood of identifying the optimal in a reasonable time frame. The simulation method is selected in consideration of the characteristics of the physical system, the time complexity of the algorithm, and the required modelling accuracy in the design problem. For example, modelling the near-field responses of nanostructure requires full wave simulation such as FEM, but predicting the diffractive intensities of grating struc-

tures simply needs a semi-analytical method such as RCWA. In a design problem, if a number of simulations are required, a discriminative model can be trained from the simulated data to avoid expensive evaluations of the device performance and accelerate the design process.

Given the data type of the photonic structures and the optical responses, various statistical and machine learning models can be selected. Specifically, in our research, the photonic structures are represented in binary images, we chose convolutional neural networks as the surrogate model for the simulation. However, simply feeding the training data into the network is problematic due to the overfitting problem. To alleviate this problem, we should always augment the data through various methods and transform the optical responses in a proper way. On the other hand, we can adapt advanced network architectures to improve the speed of convergence. We can always use the state-of-the-art deep learning models to model the photonic systems, although it does not provide any insight of the physics behind the system. When the photonic device is composed of period nanostructure, we can revise the CNN such that the periodic boundary condition is naturally satisfied throughout the network. Such a revised CNN can capture the features of interactions between adjacent cells and expedite the training process.

7.2 Encoding photonic structures with generative models

When the DOF of the photonic structure is high, traditional optimization approaches may yield some local optimal solutions or may not converge at all. Thus, we need to find an alternative encoding method such that the photonic structures can be represented in a compact, continuous space. In our case, we utilized a GAN to encoding the binary images that represent 2D metasurfaces into a low-dimensional latent space. In order to identify a nanostructure that satisfies certain spectral behavior, we modify the GAN and include three components – a generator, a critic, and a simulator. The generator produces candidates of photonic structures, while the critic and the simulator collaboratively guide the generator

to yield structures that resembles certain pattern in a geometric dataset as well as satisfies the design objective. The design process can be understood as identifying a pattern from the geometric dataset given the objective. However, different from search algorithm, the GAN-based method is able to 1) identify topologies in a much efficient way, and 2) produce patterns that are not in the geometric dataset. This is because the two essential properties of the latent space of generative models – compactness and continuity.

To validate the machine-learning-based design strategy, we chose to design single-layered metallic metasurfaces given certain transmittance requirement in the visible and near-infrared regime. We demonstrate the efficacy of the strategy for globally identifying photonic structures by several design cases. The strategy can take randomly picked, on-demand input spectra, and is able to produce multiple candidates that have similar spectral behaviors with the design objectives. The user may select one of the candidates that can be easily fabricated as the final design. This method is a global method to identify photonic structures, but, as a data-driven method, it may not accurately converge to the global optima. However, since the results are near the global optima, the optimal solution can be achieved in a few optimization iterations. In other words, this approach can be used to produce good initializations for other optimization approaches, such as adjoint methods and evolutionary algorithms.

7.3 Identifying photonic structure with optimization approaches

With a trained generative model, we can leverage the low dimensionality of its latent space, and directly perform optimization on the compact space. We trained a VAE for the generation of candidate photonic structures, the generator of which is able to transform a latent vector with a dimensionality of 10 to a binary image representing the corresponding structure. ES is implemented for the optimization of the latent vector. During the optimization, a population of latent vectors are initialized. The algorithm first recovers the vectors into the photonic structures and carries out simulations to retrieve their optical responses. After

that, the algorithm starts the standard loops of evolutionary algorithm – selection, crossover, mutation, and elimination. After one of the structures satisfies the design goal or the maximum iterations of the algorithm is reached, the optimization loop stops. We test this design strategy with the design of meta-structure given various transmittance requirement. The designed results tiny difference with respect to the design goal. It is worth noting that, with a NN simulator, the overall optimization process can be reduced to seconds. The fast optimization speed can accelerate the design cycle when various designs with same the configuration are required.

With the idea of optimizing latent vectors to identify optimal structures, we extended the framework described above with more a sophisticated generator and ES algorithm. We first replace the generator from a standard CNN to a CPPN. The CPPN composes binary patterns one pixel at a time based on the input coordinate. By manipulating the coordinates fed to the CPPN, we are able to arbitrarily control and transform the output topology. This unique feature allows us to augment the generator with increased capacity of generating patterns that are not seen in the training dataset. For the ES algorithm, we replaced it with a CC. The CC divides a large problem into subtasks, and individually solve the subtasks in a greedy way. Leveraging this property, we can design multiple meta-atoms in a meta-molecule without expensive, large-scale simulations. In a meta-molecule, we assume no strong correlation happens between two adjacent meta-atoms, and the overall performance of the meta-molecule is a linear combination of the meta-atoms. An example of such a meta-molecule is the gradient metasurface. We thus can utilize CC to separately design the meta-atoms in the molecule in a greedy way. In doing so, we can simply carry out the simulations of the meta-atoms and avoid the expensive simulation of the meta-molecules.

We applied the developed framework to the design of meta-molecules for polarization conversion and diffraction manipulation. The algorithm is able to design the diatonic meta-molecules that converts the incident LP light into LP light with arbitrary polarization angles and CP light with low DOLP for various wavelengths. We further utilize the framework to

design gradient metasurfaces for LP and CP incident light. It is surprising to see that some of the designed CP gradient metasurfaces only consist of spatially twisted nanoparticles, which is the causes of geometric phase gradient. If we do not enforce the CP metasurfaces to have the same responses to LCP and RCP light, it is possible to see the design shows distinct behaviors with incident light of opposite helicities. The designed diatonic and gradient metasurfaces are experimentally validated, and we observe accurate polarization conversion and scattering direction as predicted by simulations. In addition to the design of meta-atoms, we also implemented our methods in the modelling and design of FinFET SRAM cells [154, 155, 156, 157].

7.4 Other sparse representation of photonic structures

In fact, in some situations of data-driven inverse design, photonic structures can be encoded through mathematical transformations. Traditionally, a 2D photonic structure can be represented as the intersection of a 3D surface and a 2D plane. This method is known as a level set method. Inspired by the level set, the sparse representation of a photonic structure can be found by encoding the 3D surface into a compact space. A vector of the low-dimensional space corresponds to a photonic structure in the original parameter space. Optimization can be performed on the compact space to identify the optimal structures. The overall proposed encoding method can be summarized as follows: 1) given a photonic structure, find a function that represents the 3D surface such that the intersection of the surface and a plane is the topology of the structure, 2) Transform the function into a compact space through FT. By reversing the encoding process, we can retrieve the photonic structure from a low-dimensional vector.

We utilized the encoding method to design DOEs with various diffraction angles intensities. Because of the FT, the encoding method intrinsically filter out redundant information during the design and only allow possible DOEs with required order of diffractions to be generated. In our 3×3 DOE design, only 9 variables are used to represent the complex

topologies of the structure, which originally have to be represented by images with pixels over 4,000. Due to the vast reduction of dimensionality, we can identify the optical DOE structures with stringent diffraction intensity requirement using a data-drive method. This encoding method can be combined with generative methods to further reduce the dimensionality of photonic structures for the analysis of dataset and the design of devices. Because of the FT, the encoding method can naturally eliminate fine features that are not suitable for fabrication. In the future, we can also try to incorporate the encoding method into other optimization approaches to increase the likelihood of identifying the global optimal solution for certain physical problems.

7.5 Difficulties and challenges

Over the past few years, machine learning has successfully demonstrated its potential to yield complex high-performance photonic designs with little human intervention. However, an accurate model inevitably requires a huge amount of training data, which may incur a substantial computational burden. The tradeoff between the size of the dataset and the accuracy of the model is indeed a crucial factor to be considered when machine learning is utilized for inverse design. There are several approaches being investigated to mitigate the dependency on data. For example, using advanced machine learning techniques in combination with physical methods can increase the explainability of the model and thus improve the prediction accuracy with less data [141]. Meanwhile, deep learning has proved to be able to capture the fundamental law behind the complex physical phenomena [158], and serve as the intermediate steps for solving numerical partial differential equations [159, 160]. Such strategies provide a fast, generalized, and accurate modeling method for potential optimization and inverse problems with much lower data requirement. On the other hand, since collecting data does not conflict with traditional optimization strategies, data generation and optimization can be performed in parallel. The collected data can be used to train a machine learning model for the acceleration of the simulation in the later opti-

mization steps. Lastly, as other data-driven research, an effective approach to accelerate the development of machine-learning-assisted photonic design is to collectively construct large datasets of various optical designs with the effort of the optical community. The established dataset could avoid repeated efforts of generating simulation data, shorten the cycle of implementing new algorithms, and provide a unified standard to evaluate model performance.

7.6 Outlook

The prosperity of machine learning and artificial intelligence is bringing the scientific community onto a new stage. In this era, the analytical methods, optimization algorithms, and data-driven approaches are consolidated, forming a toolbox to uncover the theory behind complex phenomena and design unconventional devices that could never be discovered before. In the realm of optics and photonics, we have witnessed the evolution of research methodologies from analytically solving governing equations to today's AI strategies with learning and optimization. Machine learning algorithms have helped unearth the intrinsic relations between matter and light behavior, providing insights to enable and assist the design of optical components. Notwithstanding the fact that these are preliminary research, we can expect, when state-of-the-art machine learning algorithms are progressively adapted to the optical community, the complexity of the design to continue growing, and the performance to further approach the limit bounded by physics. Nowadays, due to the demanding geometric and physical restrictions, the design of optical components in modern applications have heavily relied on computation and optimization algorithms. With the union of AI and machine learning, we are anticipating machine intelligence, jointly with traditional methods, to substantially boost the discovery and development of advanced photonic devices in essential and/or unconventional applications, including optical communications, high-resolution displays, virtual/augmented reality, various sensing technologies, and so much more.

Appendices

APPENDIX A

PROBABILISTIC MODELING OF THE GENERATIVE DESIGN STRATEGY

In this chapter, we will present a probabilistic modeling of the inverse design strategies based on generative models. We prove that, under certain conditions and assumptions, the sampling-based generative approaches are able to identify the globally optimal solutions. We further illustrate the necessity of using GAN in the design of high-dimensional complex photonic structures. This chapter primally provides the probabilistic explanation of the effectiveness of the method presented in Chapter 3, and it sheds light on the potential of overall generative model-based design approaches.

A.1 Probabilistic design model

In our model, we define the input of the physical systems as x_{in} and the parameters of the optical systems as x_p. However, since the variation of the two parameters both affect the output of the system y, we unify them as a single vector x. The simulation process can be denoted as $y = F(x)$, where F is a deterministic function. We represent the design objective as y_{obj}, and thus our probability of achieving the y_{obj} given x can be modelled as a Gaussian distribution centered at y_{obj}:

$$p(y_{obj}|x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(y - y_{obj})^2\right) \tag{A.1}$$

Here σ is a scaling factor that does not affect the optimization results. Typically, the logarithm of the probability is more convenient to work with:

$$\begin{aligned} L &= \log p(y_{obj}|x) \\ &= -\log\sqrt{2\pi}\sigma - \frac{1}{2\sigma^2}\|y(x) - y_{obj}\|^2, \end{aligned} \tag{A.2}$$

In a latter discussion, we will see such a model can be interpreted to the MSE loss function in the deterministic optimization. The definition of a MSE loss function in our latter discussion is

$$L_{MSE} = \frac{1}{2}\|y(x) - y_{obj}\|^2. \tag{A.3}$$

A.2 Gradient-based optimization

The gradient-based optimization seeks the gradient of the loss function with respect to the design objective $\nabla_x L_{MSE}$, and subtracts the gradient from x iteratively until convergence. In both adjoint method and deep learning approaches, the gradient of the output with respect to the input $\partial y/\partial x$ can be explicitly calculated from the model F either by solving partial differential equations or through backpropagations. Hence, the gradient of the MSE loss function with respect to the input x amounts to

$$\nabla_x L_{MSE} = (y - y_{obj})\frac{\partial y}{\partial x} \tag{A.4}$$

Interestingly, the gradient of the log probability (Eq. A.2) is

$$\nabla_x L = -\frac{1}{\sigma^2}(y - y_{obj})\frac{\partial y}{\partial x} \tag{A.5}$$

Form the above equations, we can conclude that the minimize MSE loss is equivalent to maximizing the log probability $\log p(y_{obj}|x)$. Thus, the gradient method can be interpreted as identifying the parameters x such that the probability of observing the desired y_{obj} is maximized.

A.3 Sampling-based design strategy

Now, we consider sampling multiple designs x^i and optimize the overall performance of all the samples. We denote the optical responses and the probability of achieving the design

objectives as y^i and y^i_{obj}, respectively. Assuming each design is independent, we thus can calculate the probability of all samples $p(y_{all})$ as:

$$p(y_{all}) = \prod_i p(y^i_{obj}) \tag{A.6}$$

The log likelihood of $p(y_{all})$ has the form

$$L_{smpl} = -N \log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2} \sum_i \|y^i - y^i_{obj}\|^2 \tag{A.7}$$

where N is the sample size. In order to maximize the log probability, we compute the gradient:

$$\nabla_x L_{smpl} = -\sum_i \nabla_x L_i \tag{A.8}$$

where $L_i = \|y^i - y^i_{obj}\|^2$. Equation A.8 indicates that the sampling algorithm takes into account all the design errors. Hence, the designed structure can be more robust compared to the designs through deterministic optimization schemes. Indeed, such a sampling approach is able to identify global solution – the optimal solution maximizes $p(y_{obj})$ instead of $p(y_{obj}|x)$. In order to see that, we expand the probability of achieving the objective $p(y_{obj})$ with respect to the design parameters x:

$$\begin{aligned} p(y_{obj}) &= \int p(y_{obj}, x) dx \\ &= \int p(y_{obj}|x) p(x) dx \\ &= \mathbb{E}_x \left[p(y_{obj}|x) \right] \end{aligned} \tag{A.9}$$

The log probability of $p(y_{obj})$ is

$$\begin{aligned}\log p(y_{obj}) &= \log \mathbb{E}_x \left[p(y_{obj}|x)\right] \\ &\geq \mathbb{E}_x \left[\log p(y_{obj}|x)\right] \\ &= \mathbb{E}_z \left[-\log \sqrt{2\pi}\sigma - \frac{1}{2\sigma^2}\|y - y_{obj}\|^2\right] \\ &\propto L_{smpl}\end{aligned} \tag{A.10}$$

We used Jensen's inequality in the above equation. From Eq. A.10, we can see that maximizing L_{smpl} is equivalent to maximizing the lower bound of $\log p(y_{obj})$. Therefore, we can conclude that optimizing through the sampling algorithm is more likely to converge to a global solution than deterministic optimization methods.

A.4 Generative design strategy

Although the sampling algorithm is able to identify global optima, it barely yields a satisfiable solution in practice. This is because the geometric parameter x is not a good representation due to the roughness of the design space and rareness of the local/global optima, and thus, it is challenging to locate the only optima in the original x representation. Therefore, we seek to reparametrize x through another smooth function with much more parameters w and optimize the newly introduced parameters. In doing do, we automatically create multiple sets of w that could possibly represent good candidates of x. Identifying any set of w is more feasible than locating the only few optima in the original x representation. Optimizing w does not incur additional expensive computation if we use automatic differentiation.

A.4.1 The generative model

In detail, we take a step further and consider the generative model:

$$x = G_w(z) \tag{A.11}$$

where G_w is a function parametrized by w and maps a random vector z to a vector x in the design space. We can assume z is a sample from a distribution $f_Z(z)$. In the generative approach, instead of directly optimizing x, we choose to optimize w such that the generated x has the desired optical responses. Since the generator is a deterministic function, we have

$$p(x|z) = \delta(x - G_w(z)) \tag{A.12}$$

We can find derive the distribution $p(x)$ through

$$p(x) = \int p(x|z)p(z)dz \tag{A.13}$$

If the function $G_w(z)$ is invertible and G_w^{-1} is differentiable, we can explicitly derive

$$p(x) = f_z(z) \left| \frac{\partial z}{\partial x} \right| \tag{A.14}$$

In practice, the $p(x)$ may not have a closed form, but fortunately, we can optimize L_{smpl} without knowing the $p(x)$. Using the chain rule, the gradient of the L_{smpl} with respect to w is:

$$\nabla_w L_{smpl} = -\sum_i \nabla_x L_i \frac{\partial z}{\partial x} \tag{A.15}$$

L_{smpl} can be improved by iteratively subtracting the gradient from it. After optimizing w, the G_w is able to produce population of samples that satisfies the design objectives. The best individual can be selected as the final design.

Additionally, we can also maximize the distribution $p(y_{obj})$ using the similar approach. The gradient of $p(y_{obj})$ with respective to the parameters w is

$$\nabla_w \, p(y_{obj}) = \mathbb{E}_z \left[\nabla_y \, p(y_{obj}|x) \frac{\partial y}{\partial x} \frac{\partial x}{\partial w} \right] \tag{A.16}$$

Using the identity

$$\nabla_y \log p(y) = \frac{\nabla_y p(y)}{p(y)} \tag{A.17}$$

and simplifying the equation, we can rewrite the gradient as

$$\nabla_w p(y_{obj}) = -\mathbb{E}_z \left[p(y_{obj}|x)(y - y_{obj}) \frac{\partial y}{\partial x} \frac{\partial x}{\partial w} \right] \tag{A.18}$$

Compare this with the gradient of MSE loss:

$$\nabla_w L_{MSE} \propto \mathbb{E}_z \left[(y - y_{obj}) \frac{\partial y}{\partial x} \frac{\partial x}{\partial w} \right] \tag{A.19}$$

Regardless of the signs, they are off by $p(y_{obj}|x)$. Compared to directly minimizing MSE loss (Eq. A.19), maximizing $p(y_{obj})$ (Eq. A.18) also considers the "quality" of each sample (indicated by $p(y_{obj}|x)$.) and thus is superior. However, if $p(y_{obj}|x)$ is sufficiently small, especially at the beginning stage of the optimization, the gradient is diminished and the optimization is stagnant. Therefore, a gradually decreasing learning rate α with a relatively large initial value is typically applied to L_{MSE} to replace $\nabla_w p(y_{obj})$ for a better performance. Note Eq. A.18 has the identical form as the loss function proposed in Ref. [130]

A.4.2 Bayesian perspective

In the above discussion, we did not consider the prior distribution of the design parameters $p(x)$. In fact, we can identify x by looking for maximizing $p(x|y_{obj})$. According to the Bayes' theorem, for each sample, we can rewrite $p(x^i|y^i_{obj})$ as

$$\begin{aligned} p(x^i|y^i_{obj}) &= \frac{p(y^i_{obj}|x^i)p(x^i)}{p(y^i_{obj})} \\ &\propto p(y^i_{obj}|x^i)p(x^i) \end{aligned} \tag{A.20}$$

In the following discussion, we will neglect the constant term $p(y^i_{obj})$. Assuming each

sample is independent, we can have the log likelihood of all samples as

$$L' = \log \prod_i p(x^i | y^i_{obj}) \tag{A.21}$$

Expanding L' we have

$$\begin{aligned} L' &= \sum_i \log p(y^i_{obj} | x^i) + \sum_i \log p(x^i) \\ &= -N L_{MSE} + \sum_i \log p(x^i) \end{aligned} \tag{A.22}$$

If the explicit form of $p(x^i)$ can be found using Eq. A.14, the second term can be rewritten as

$$\log p(x^i) = \log f_Z(z) + \log \left| \frac{\partial}{\partial x} G_w^{-1}(x^i) \right| \tag{A.23}$$

Thus,

$$L' = -N L_{MSE} + \sum_i \log f_Z(z^i) + \sum_i \log \left| \frac{\partial}{\partial x} G_w^{-1}(x^i) \right| \tag{A.24}$$

The gradient with respect to w amounts to

$$\nabla_w L' = -N \nabla_w L_{MSE} + \sum_i \frac{\partial}{\partial w} \log \left| \frac{\partial}{\partial x} G_w^{-1}(x^i) \right| \tag{A.25}$$

Eq. A.25 contains an additional term and essentially tries to maximize the $p(x)$ through sampling. However, such a method does not provide a practical algorithm. This is because 1) the inverse of $G_w(x)$ may not exist, and 2) the additional term requires massive sampling to be accurately estimated, which is impossible for problems with high dimensional design space. In a later section, we will illustrate the additional term can be used to expedite the optimization convergence by enforcing the distribution of the samples to have a similar distribution as a predefined dataset.

A.5 Knowing the existing data distribution

A.5.1 Importance sampling

In practice, if the design space is high dimensional, the sampling algorithm may suffer from high variance and failed to converge. But if we already know a prior distribution of the design parameters $q(x)$, we can approximate the likelihood in a faster manner. In detail, we can expand $p(y_{obj})$ to include $q(x)$ as

$$\begin{aligned} p(y_{obj}) &= \int p(y_{obj}|x)\frac{p(x)}{q(x)}q(x)dx \\ &= \mathbb{E}_{x\sim q(x)}\left[p(y_{obj}|x)\frac{p(x)}{q(x)}\right] \end{aligned} \tag{A.26}$$

However, invertibility of the function G_w is essential for the implementation of importance sampling. Fortunately, it is possible to construct an invertible neural network [161]. The very first step of the procedure is to derive the distribution $q(x)$ from the data. With the constructed distribution, we can perform the efficient sampling. In each iteration of the sampling, we first sample $x \sim q(x)$, and derive the latent vectors z through the inverse function $z = G_w^{-1}(x)$. The PDF $p(x)$ can be directly computed through Eq. A.14.

A.5.2 Distribution distances

In addition to importance sampling, we can enforce the distribution of $p(x)$ to be similar to the prior distribution $q(x)$ as discussed in section A.4.2. For example, we add an additional term that minimizes the Jensen–Shannon divergence (JS) of the two distribution into the loss function. The revised loss function can be written as

$$L_G = L_{MSE} + \lambda D_{JS}(p(x)\|q(x)) \tag{A.27}$$

Expand $D_{JS}(p(x)\|q(x))$ using identity

$$D_{JS}(p\|q) = D_{KL}\left(p\|\frac{p+q}{2}\right) + D_KL\left(q\|\frac{p+q}{2}\right) \tag{A.28}$$

We have

$$L_G = L_{MSE} + \lambda \int p \log \frac{p}{p+q} dx + \int q \log \frac{q}{p+q} dx + const. \tag{A.29}$$

Define function

$$D(x) = \frac{q(x)}{p(x)+q(x)} \tag{A.30}$$

Equation A.29 changes to

$$\begin{aligned} L_G &= L_{MSE} + \lambda \int p \log(1 - D(x)) + q \log D(x) dx + const. \\ &= L_{MSE} + \lambda \mathbb{E}_{x\sim p(x)}\left[1 - D(x)\right] + \lambda \mathbb{E}_{x\sim q(x)}\left[D(x)\right] + const. \end{aligned} \tag{A.31}$$

In order to optimize the generative model, we calculate the gradient of L_G:

$$\begin{aligned} \nabla_w L_G &= \nabla_w L_{MSE} - \lambda \mathbb{E}_{x\sim p(x)}\left[\nabla_w D(x)\right] \\ &= \nabla_w L_{MSE} - \lambda \mathbb{E}_{z\sim p(z)}\left[\frac{\partial D(x)}{\partial x}\frac{\partial x}{\partial w}\right] \end{aligned} \tag{A.32}$$

With the gradient of L_G, we can minimize the MSE loss and enforce the generated samples to have a similar distribution with the prior data. It is noteworthy that other probabilistic divergence can be used as the penalty term of the loss function.

A.6 Generative adversarial network

The methods described in section A.4 can be used when $q(x)$ is relatively easy to retrieve from the prior data. In most of the cases, the $q(x)$ cannot be well approximated with limited amount of data. Hence, we need to learn the distribution $q(x)$ while performing the

optimization. Here we only consider the optimization method illustrated in A.5.2. In fact, we can parametrize the function $D(x)$ directly since $q(x)$ is included in D. Let $D_\theta(x)$ as a function parametrized by θ, then another aim during the optimization is to learn an optimal θ such that the distance between $p(x)$ and $q(x)$ can be measured. From the definition of $D_\theta(x)$, we know that if two distributions are indistinguishable, we have $D_\theta(x) = 1/2$, otherwise the value of it approaches 0 or 1. This is the fundamental idea of a GAN. $D_\theta(x)$ in this case is the discriminator/critic. The loss function of the D is

$$L_D = \mathbb{E}_{x\sim p(x)}\left[1 - D_\theta(x)\right] + \mathbb{E}_{x\sim q(x)}\left[D_\theta(x)\right] \tag{A.33}$$

Combined with the loss of the generator (Eq. A.31), we are able to approximate the prior distribution $q(x)$ while optimize the parameters x as in the training of GAN. The detailed implementation is the topic presented in Chapter 3.

APPENDIX B

MANIPULATING PHOTONIC STRUCTURES WITH CPPN

In this section, we provide additional information and proproteins of CPPN that can be useful for encoding photonic structures and generating complex topologies. Since CPPN construct image pixel by pixel, we are able to vary the generated pattern by transforming input coordinate. This unique property provides tremendous degree of freedom for controlling the complicated photonic structures in various applications.

We first formulate the network into a mathematical function. We denote (x, y) as coordinates in the image space, $r = \sqrt{(x^2 + y^2)}$ as the radial distance of the point (x, y) to the center of the image, v as a random vector with arbitrary dimension, and N as the number of pixels of the generated image. For a CPPN network G_θ parametrized by weights θ, the output of the CPPN is:

$$u_i = G_\theta(x_i, y_i, r_i, v) \tag{B.1}$$

where $i \in N$ is the index of the pixel coordinates. The full image is composed by looping all the coordinates tuple (x, y, r) in the image space and thus we denote the generated image as the $\{G_\theta(x_i, y_i, r_i, v) \ | i \in N\}$. To simplify the notation, we denote $X = \{x_i | i \in N\}$ as the set of all x coordinates and adopt the similar notations for Y and R, then the input of the CPPN is the product space $X \times Y$. The composed image thus can be written in a compact form:

$$u = G_\theta(X \times Y, R, v) \tag{B.2}$$

From this equation, we can observer that the generated pattern is associated with both the bias vector v and the input coordinates. We can expect to transform the shape of composed

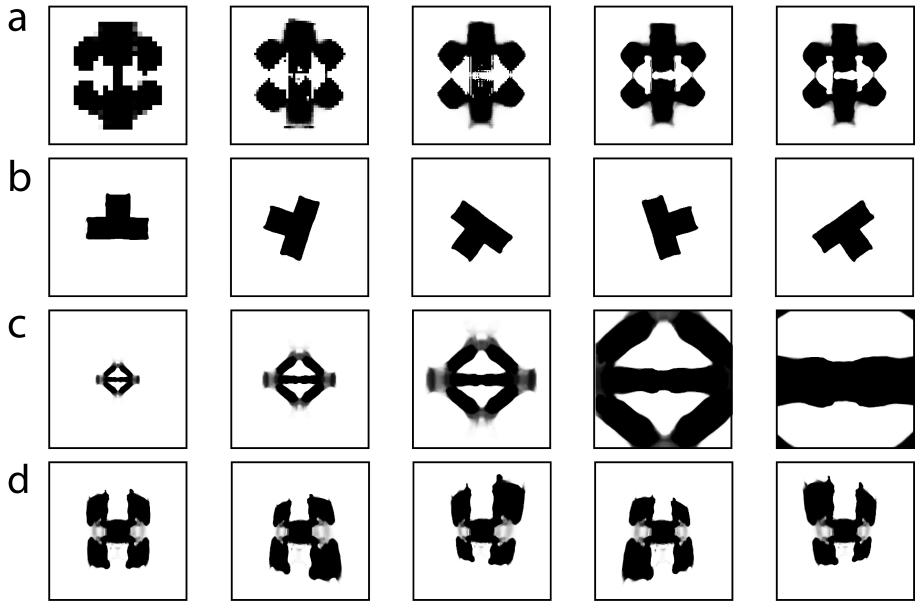


Figure B.1: Transforms on the composed patterns. (a) Composed patterns from the CPPN with the same bias vector v but in various resolutions. It should be noted that the CNNP is only trained with patterns in a resolution of 64×64. (b) Rotating the same patterns with various angles. (c) Performing affine transformations to the same pattern. Here only scaling transformations are presented. (d) Performing perspective transformation to the same pattern.

image simply by transforming the input coordinates to X', Y', and R'. A few transformations and their examples are discussed as follows.

a. Changing the resolution of the composed images

The CPPN is trained with geometric data with dimension of 64×64. However, as we can interpolate the input coordinate, the generated images can be in arbitrary dimension. Figure B.1(a) displays the examples when we alter the number of points in the set of X', Y', and R' from 32 to 512. The increased resolution does not affect the shape of the images, while the edges of the patterns are smoother. This property allows us to design photonic devices with arbitrary dimension.

b. Rotating the images

We can also perform global transformations like rotation and shift to the composed images by doing the same transformation to the input coordinates. Suppose we need to rotate the pattern with an angle α, the rotation matrix can be written as:

$$M(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \tag{B.3}$$

The transformed coordinates are $[X', Y']^T = M(\alpha)[X, Y]^T$. Examples of rotating various angles from 0 to 2π are shown in Figure B.1(b).

c. Affine and perspective transformations

The global linear transformation of an image can be done by affine transformation or perspective transformation. The matrix for the affine transformation is:

$$M_{aff} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{B.4}$$

where entries a,b,c,and d determines the shape of the transformed pattern and t_x,t_y controls the translation of the pattern. The perspective transformation takes a similar form:

$$M_{aff} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ e & f & 1 \end{bmatrix} \tag{B.5}$$

The additional e and f parameters endow the richness of transformed patterns. Given input

coordinates $[X, Y]$, the transformation follows the manner:

$$\begin{bmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{W} \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{B.6}$$

and

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \frac{1}{\|\tilde{W}\|} \begin{bmatrix} \tilde{X} \\ \tilde{Y} \end{bmatrix} \tag{B.7}$$

These two transformation approaches cover most of the basic transformations needed in photonic design such as reflect, rotate, shear, and translate. A few examples of transformed patterns using the affine and perspective transformations are displayed in Figure B.1(c) and (d).

d. Other transformations

In addition to the aforementioned global transformations, it is feasible to locally transform patterns by applying different transformations at different regions of the images. Relying on the flexibility of the coordinate transformation, we can enforce symmetry properties and the connectedness of the pattern without losing the gradients. These local transformations may be used in the inverse design of photonic structures with high symmetric topologies such as those in the lattices of photonic crystals.

APPENDIX C

ADDITIONAL EXAMPLES OF DESIGNED META-MOLECULES

In this section, we provide additional examples of meta-molecule design as in Chapter 5. Figure C.1(a), (d), (g), and (j) displays the diatonic metasurfaces function at wavelengths of $1765\,\mathrm{nm}$, $1111\,\mathrm{nm}$, $938\,\mathrm{nm}$, $638\,\mathrm{nm}$. The aimed rotation angle/conversed polarization are indicated at the bottom of each design. Figure C.1(b), (e), (h), and (k) are NN-simulated polarization states of the transmitted light, while Figure C.1(c), (f), (i), and (l) are FEM-simulated states. The states in the polar plots are normalized. Figure C.2, Figure C.3 show the designed linear gradient metasurfaces and Figure C.4, Figure C.5 display the circular gradient metasurfaces. The left column of each figure shows the eight elements in the meta-molecules, the two middle columns describe the amplitude and phase of the eight meta-atoms, and the rightmost column provides the FEM-simulated electric field at the functioning wavelengths.

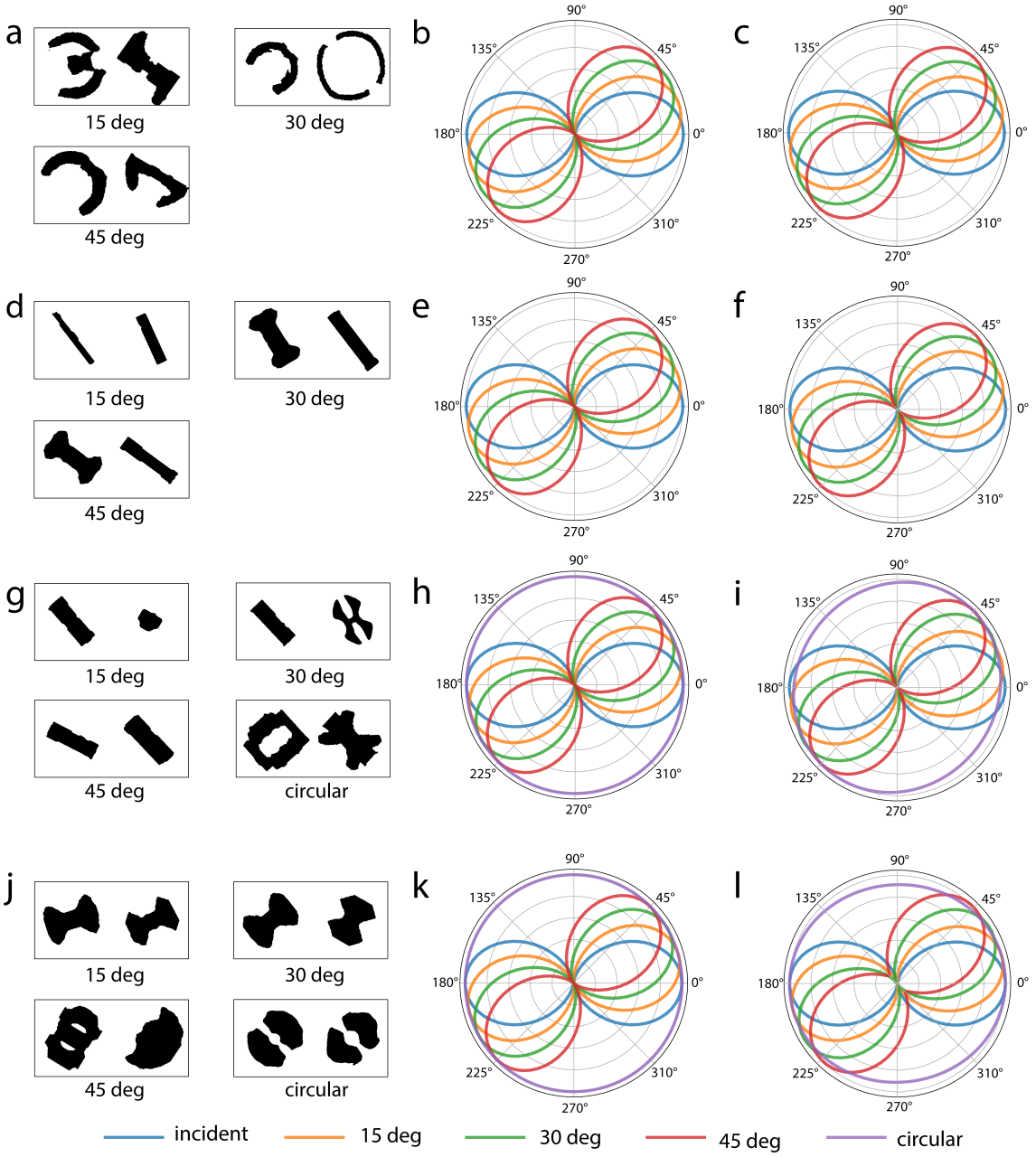


Figure C.1: More examples of diatomic metasurfaces for polarization manipulation. (a, d, g, and j) The identified unit cell of the metasurface functioning at the wavelength of 1765 nm, 1111 nm, 938 nm, 638 nm, respectively. The design objectives are prescribed polarization rotation/conversion for the transmitted wave as indicated underneath each metaphotonic structure. (b, e, h, and k) Neural network-simulated polarization state of the transmitted light. (c, f, i, and l) FEM-simulated state of polarization for the transmitted wave. The polarization states in the polar plots are normalized to their respective maximum intensity.

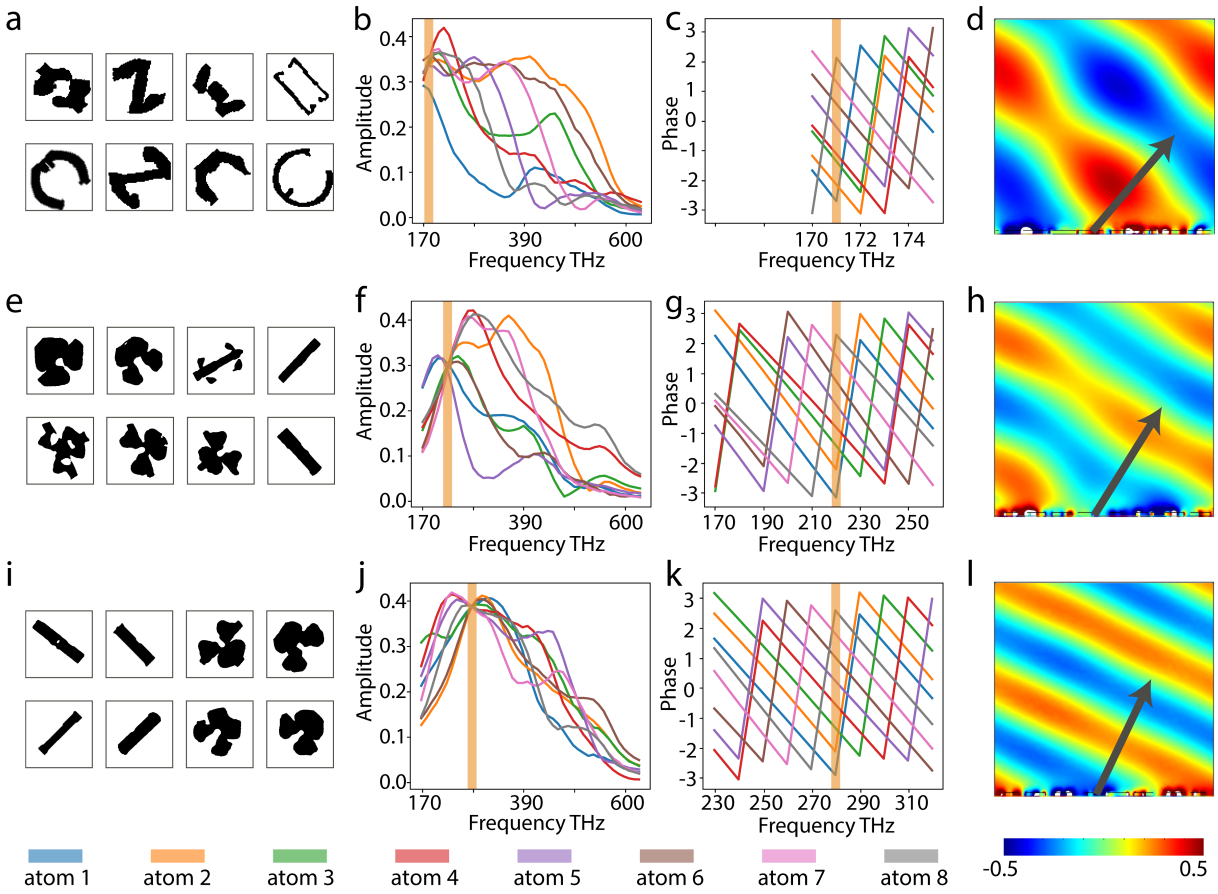


Figure C.2: More examples of gradient metasurfaces for linearly polarized light (part 1). The design objective is to convert x-polarized incident light into y-polarized and to deflect this portion of light into a specific direction. (a, e, and i) Designed unit cells of the gradient metasurfaces operating at the wavelength of $1666\,\mathrm{nm}$, $1364\,\mathrm{nm}$, $1111\,\mathrm{nm}$, respectively. (b, f, and j) The corresponding cross-polarized transmittance of the eight meta-atoms shown in the first column. The orange line indicates the operating frequency/wavelength of the design. (c, g, and k) The corresponding phase delay of the eight meta-atoms shown in the first column. (d, h, and l) Simulated distribution of the electric field E_y induced by the gradient metasurface shown in (a), (e), and (i), respectively. The solution for each working wavelength is not unique, and here we selectively present one of the designed metasurfaces.

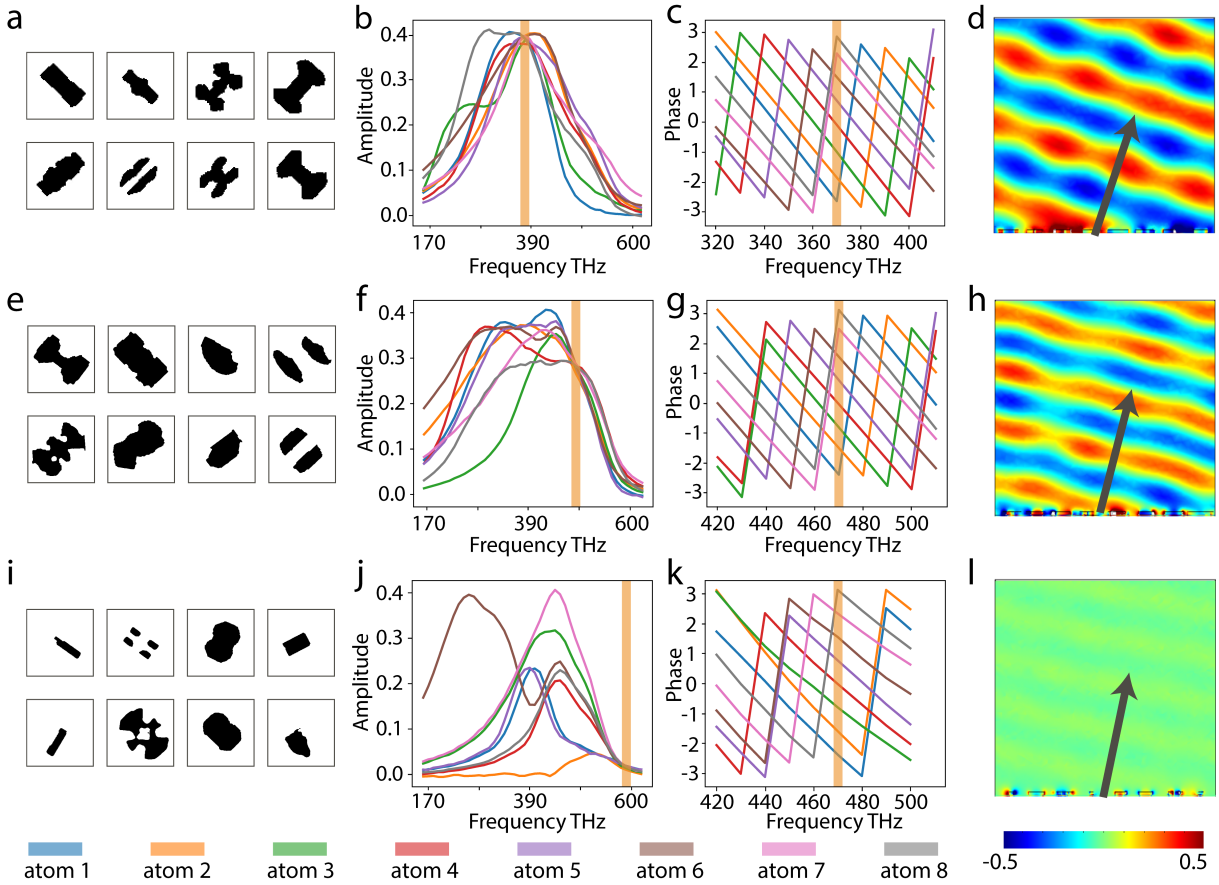


Figure C.3: More examples of gradient metasurfaces for linearly polarized light (part 2). (from top to bottom) Designed meta-molecules as in Fig. S9 at the wavelength of 810 nm, 638 nm, 512 nm, respectively.

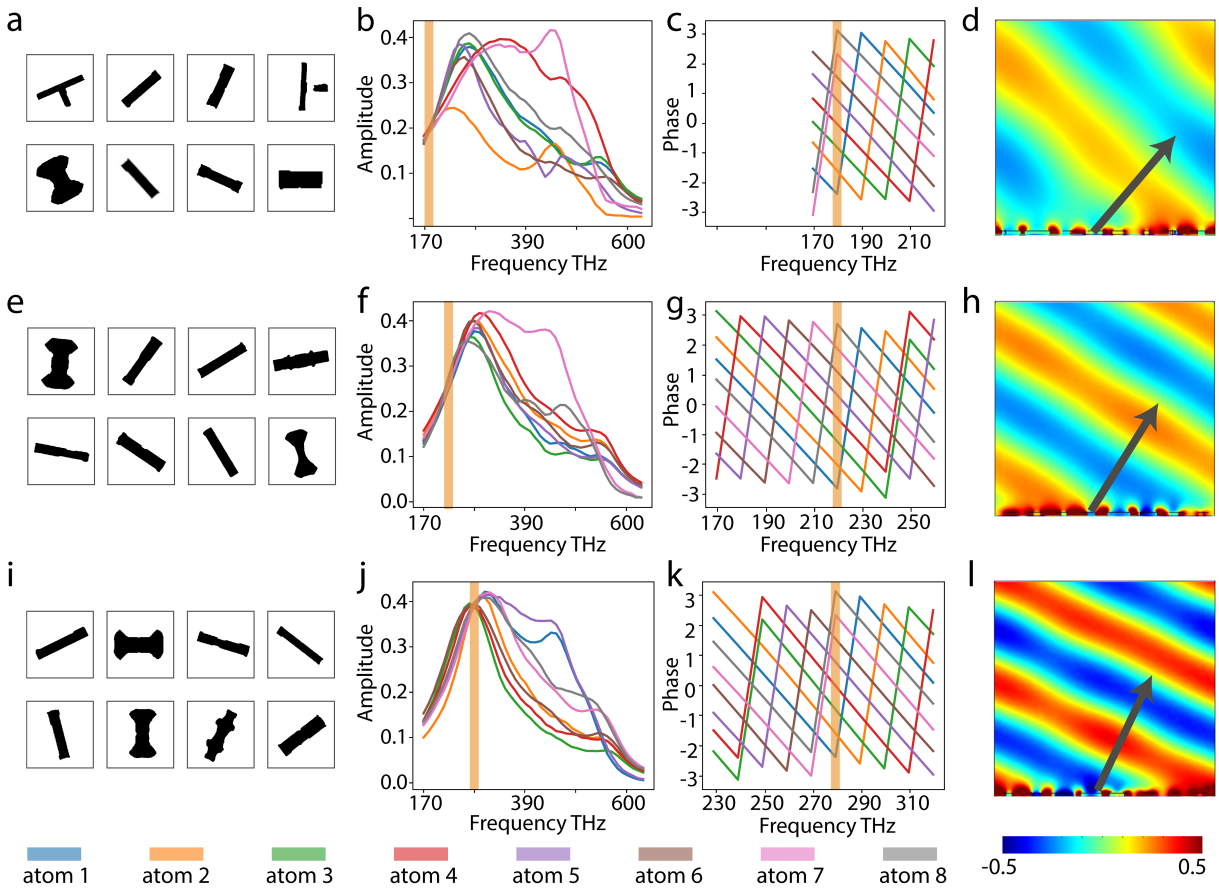


Figure C.4: More examples of gradient metasurfaces for circularly polarized light (part 1). The design objective is to convert LCP incident light into RCP and to deflect this portion of light into a specific direction. (from top to bottom) Designed meta-molecules at the wavelength of 1666 nm, 1364 nm, 1111 nm respectively.

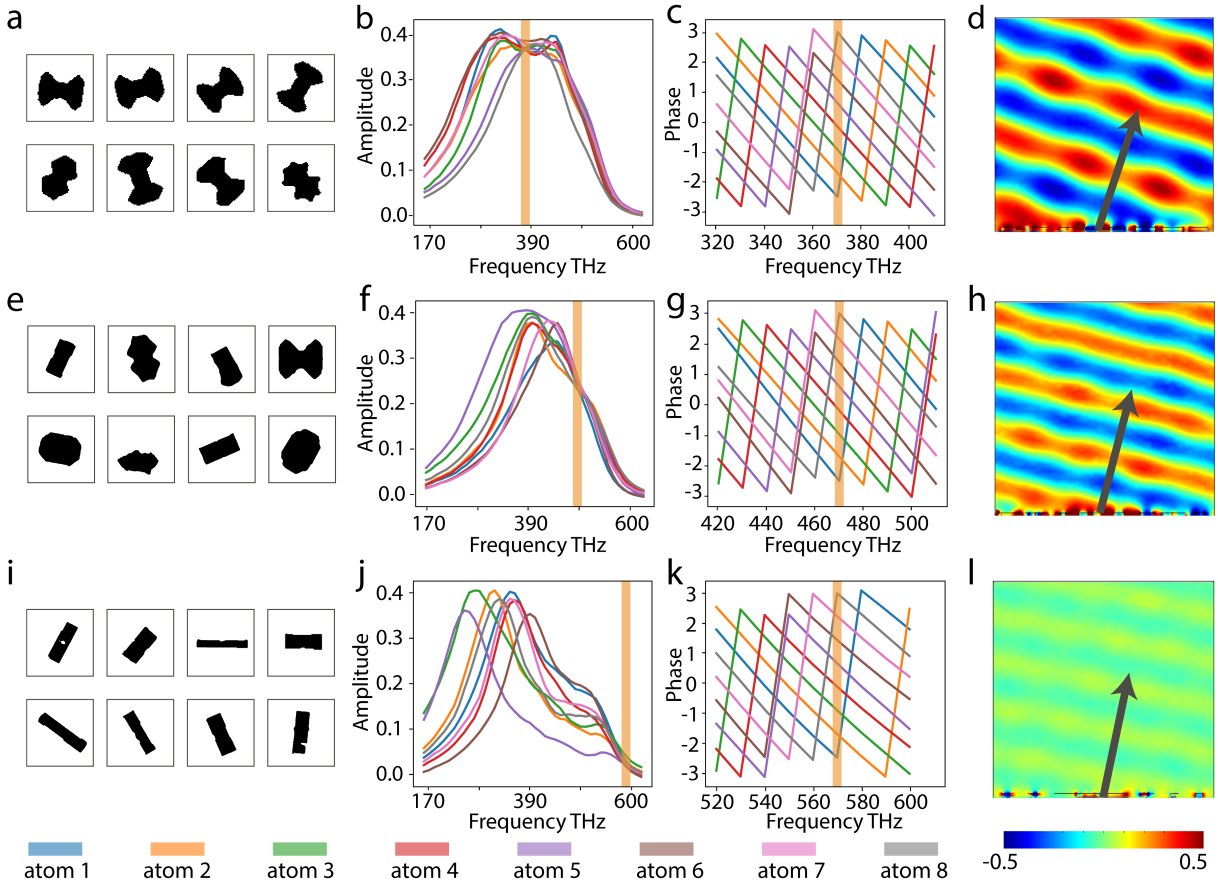


Figure C.5: More examples of gradient metasurfaces for circularly polarized light (part 2). (from top to bottom) Designed meta-molecules as in Fig. S11 at the wavelength of 810 nm, 638 nm, 512 nm, respectively.

REFERENCES

[1] W. Cai and V. M. Shalaev, *Optical metamaterials: fundamentals and applications*. Springer, 2010, ISBN: 1441911510.

[2] G. Roelkens, D. Van Thourhout, and R. Baets, "High efficiency silicon-on-insulator grating coupler based on a poly-silicon overlay," *Optics Express*, vol. 14, no. 24, pp. 11 622–11 630, 2006.

[3] D. Taillaert, W. Bogaerts, P. Bienstman, T. F. Krauss, P. Van Daele, I. Moerman, S. Verstuyft, K. De Mesel, and R. Baets, "An out-of-plane grating coupler for efficient butt-coupling between compact planar waveguides and single-mode fibers," *IEEE Journal of Quantum Electronics*, vol. 38, no. 7, pp. 949–955, 2002.

[4] K. O. Hill, Y. Fujii, D. C. Johnson, and B. S. Kawasaki, "Photosensitivity in optical fiber waveguides - application to reflection filter fabrication," *Applied Physics Letters*, vol. 32, no. 10, pp. 647–649, 1978.

[5] G. Meltz, W. W. Morey, and W. H. Glenn, "Formation of bragg gratings in optical fibers by a transverse holographic method," *Optics Letters*, vol. 14, no. 15, pp. 823–825, 1989.

[6] G. A. Ball and W. W. Morey, "Compression-tuned single-frequency bragg grating fiber laser," *Optics Letters*, vol. 19, no. 23, pp. 1979–1981, 1994.

[7] Y. J. Rao, "In-fibre bragg grating sensors," *Measurement Science and Technology*, vol. 8, no. 4, pp. 355–375, 1997.

[8] K. Aydin, V. E. Ferry, R. M. Briggs, and H. A. Atwater, "Broadband polarization-independent resonant light absorption using ultrathin plasmonic super absorbers," *Nature Communications*, vol. 2, 2011.

[9] J. Zhao and M. A. Green, "Optimized antireflection coatings for high-efficiency silicon solar-cells," *IEEE Transactions on Electron Devices*, vol. 38, no. 8, pp. 1925–1934, 1991.

[10] D. Sievenpiper, L. J. Zhang, R. F. J. Broas, N. G. Alexopolous, and E. Yablonovitch, "High-impedance electromagnetic surfaces with a forbidden frequency band," *Ieee Transactions on Microwave Theory and Techniques*, vol. 47, no. 11, pp. 2059–2074, 1999.

[11] E. Yablonovitch, T. J. Gmitter, and K. M. Leung, "Photonic band-structure - the face-centered-cubic case employing nonspherical atoms," *Physical Review Letters*, vol. 67, no. 17, pp. 2295–2298, 1991.

[12] E. Yablonovitch, "Inhibited spontaneous emission in solid-state physics and electronics," *Physical Review Letters*, vol. 58, no. 20, pp. 2059–2062, 1987.

[13] Y. Akahane, T. Asano, B. S. Song, and S. Noda, "High-q photonic nanocavity in a two-dimensional photonic crystal," *Nature*, vol. 425, no. 6961, pp. 944–947, 2003.

[14] A. Mekis, J. C. Chen, I. Kurland, S. H. Fan, P. R. Villeneuve, and J. D. Joannopoulos, "High transmission through sharp bends in photonic crystal waveguides," *Physical Review Letters*, vol. 77, no. 18, pp. 3787–3790, 1996.

[15] S. Noda, K. Tomoda, N. Yamamoto, and A. Chutinan, "Full three-dimensional photonic bandgap crystals at near-infrared wavelengths," *Science*, vol. 289, no. 5479, pp. 604–606, 2000.

[16] M. Soljacic and J. D. Joannopoulos, "Enhancement of nonlinear effects using photonic crystals," *Nature Materials*, vol. 3, no. 4, pp. 211–219, 2004.

[17] P. Russell, "Photonic crystal fibers," *Science*, vol. 299, no. 5605, pp. 358–362, 2003.

[18] S. Y. Lin, E. Chow, V. Hietala, P. R. Villeneuve, and J. D. Joannopoulos, "Experimental demonstration of guiding and bending of electromagnetic waves in a photonic crystal," *Science*, vol. 282, no. 5387, pp. 274–276, 1998.

[19] S. A. Maier, *Plasmonics: Fundamentals and Applications*. Springer US, 2010, ISBN: 9781441941138.

[20] S. A. Maier, "The benefits of darkness," *Nature Materials*, vol. 8, no. 9, pp. 699–700, 2009.

[21] W. L. Barnes, A. Dereux, and T. W. Ebbesen, "Surface plasmon subwavelength optics," *Nature*, vol. 424, no. 6950, pp. 824–830, 2003.

[22] N. Liu, L. Langguth, T. Weiss, J. Kastel, M. Fleischhauer, T. Pfau, and H. Giessen, "Plasmonic analogue of electromagnetically induced transparency at the drude damping limit," *Nature Materials*, vol. 8, no. 9, pp. 758–762, 2009.

[23] T. W. Ebbesen, H. J. Lezec, H. F. Ghaemi, T. Thio, and P. A. Wolff, "Extraordinary optical transmission through sub-wavelength hole arrays," *Nature*, vol. 391, no. 6668, pp. 667–669, 1998.

[24] M. E. Stewart, C. R. Anderton, L. B. Thompson, J. Maria, S. K. Gray, J. A. Rogers, and R. G. Nuzzo, "Nanostructured plasmonic sensors," *Chemical Reviews*, vol. 108, no. 2, pp. 494–521, 2008.

[25] J. A. Schuller, E. S. Barnard, W. S. Cai, Y. C. Jun, J. S. White, and M. L. Brongersma, "Plasmonics for extreme light concentration and manipulation," *Nature Materials*, vol. 9, no. 3, pp. 193–204, 2010.

[26] Z. Liu, S. Chen, J. Li, H. Cheng, Z. Li, W. Liu, P. Yu, J. Xia, and J. Tian, "Fully interferometric controllable anomalous refraction efficiency using cross modulation with plasmonic metasurfaces," *Optics Letters*, vol. 39, no. 23, pp. 6763–6766, 2014.

[27] B. Luk'yanchuk, N. I. Zheludev, S. A. Maier, N. J. Halas, P. Nordlander, H. Giessen, and C. T. Chong, "The fano resonance in plasmonic nanostructures and metamaterials," *Nature Materials*, vol. 9, no. 9, pp. 707–715, 2010.

[28] Y. M. Liu and X. Zhang, "Metamaterials: A new frontier of science and technology," *Chemical Society Reviews*, vol. 40, no. 5, pp. 2494–2507, 2011.

[29] V. M. Shalaev, "Optical negative-index metamaterials," *Nature Photonics*, vol. 1, no. 1, pp. 41–48, 2007.

[30] J. Valentine, S. Zhang, T. Zentgraf, E. Ulin-Avila, D. A. Genov, G. Bartal, and X. Zhang, "Three-dimensional optical metamaterial with a negative refractive index," *Nature*, vol. 455, no. 7211, 376–U32, 2008.

[31] V. M. Shalaev, W. S. Cai, U. K. Chettiar, H. K. Yuan, A. K. Sarychev, V. P. Drachev, and A. V. Kildishev, "Negative index of refraction in optical metamaterials," *Optics Letters*, vol. 30, no. 24, pp. 3356–3358, 2005.

[32] D. R. Smith, J. B. Pendry, and M. C. K. Wiltshire, "Metamaterials and negative refractive index," *Science*, vol. 305, no. 5685, pp. 788–792, 2004.

[33] P. Moitra, Y. M. Yang, Z. Anderson, I. I. Kravchenko, D. P. Briggs, and J. Valentine, "Realization of an all-dielectric zero-index optical metamaterial," *Nature Photonics*, vol. 7, no. 10, pp. 791–795, 2013.

[34] N. I. Zheludev and Y. S. Kivshar, "From metamaterials to metadevices," *Nature Materials*, vol. 11, no. 11, pp. 917–924, 2012.

[35] R. Liu, C. Ji, J. J. Mock, J. Y. Chin, T. J. Cui, and D. R. Smith, "Broadband ground-plane cloak," *Science*, vol. 323, no. 5912, pp. 366–369, 2009.

[36] J. B. Pendry, D. Schurig, and D. R. Smith, "Controlling electromagnetic fields," *Science*, vol. 312, no. 5781, pp. 1780–1782, 2006.

[37] N. M. Estakhri and A. Alu, "Recent progress in gradient metasurfaces," *Journal of the Optical Society of America B-Optical Physics*, vol. 33, no. 2, A21–A30, 2016.

[38] X. J. Ni, N. K. Emani, A. V. Kildishev, A. Boltasseva, and V. M. Shalaev, "Broadband light bending with plasmonic nanoantennas," *Science*, vol. 335, no. 6067, pp. 427–427, 2012.

[39] N. F. Yu, P. Genevet, M. A. Kats, F. Aieta, J. P. Tetienne, F. Capasso, and Z. Gaburro, "Light propagation with phase discontinuities: Generalized laws of reflection and refraction," *Science*, vol. 334, no. 6054, pp. 333–337, 2011.

[40] H. Cheng, Z. Liu, S. Chen, and J. Tian, "Emergent functionality and controllability in few-layer metasurfaces," *Advanced Materials*, vol. 27, no. 36, pp. 5410–5421, 2015.

[41] Z. Liu, Z. Li, Z. Liu, J. Li, H. Cheng, P. Yu, W. Liu, C. Tang, C. Gu, J. Li, *et al.*, "High-performance broadband circularly polarized beam deflector by mirror effect of multinanorod metasurfaces," *Advanced Functional Materials*, vol. 25, no. 34, pp. 5428–5434, 2015.

[42] Z. Liu, Z. Li, Z. Liu, H. Cheng, W. Liu, C. Tang, C. Gu, J. Li, H.-T. Chen, S. Chen, *et al.*, "Single-layer plasmonic metasurface half-wave plates with wavelength-independent polarization conversion angle," *ACS Photonics*, vol. 4, no. 8, pp. 2061–2069, 2017.

[43] G. X. Zheng, H. Muhlenbernd, M. Kenney, G. X. Li, T. Zentgraf, and S. Zhang, "Metasurface holograms reaching 80% efficiency," *Nature Nanotechnology*, vol. 10, no. 4, pp. 308–312, 2015.

[44] A. Arbabi, Y. Horie, M. Bagheri, and A. Faraon, "Dielectric metasurfaces for complete control of phase and polarization with subwavelength spatial resolution and high transmission," *Nature Nanotechnology*, vol. 10, no. 11, 937–U190, 2015.

[45] F. Aieta, M. A. Kats, P. Genevet, and F. Capasso, "Multiwavelength achromatic metasurfaces by dispersive phase compensation," *Science*, vol. 347, no. 6228, pp. 1342–1345, 2015.

[46] N. F. Yu and F. Capasso, "Flat optics with designer metasurfaces," *Nature Materials*, vol. 13, no. 2, pp. 139–150, 2014.

[47] S. Lan, X. Zhang, M. Taghinejad, S. Rodrigues, K.-T. Lee, Z. Liu, and W. Cai, "Metasurfaces for near-eye augmented reality," *ACS Photonics*, vol. 6, no. 4, pp. 864–870, 2019.

[48] Z. Li, H. Cheng, Z. Liu, S. Chen, and J. Tian, "Plasmonic airy beam generation by both phase and amplitude modulation with metasurfaces," *Advanced Optical Materials*, vol. 4, no. 8, pp. 1230–1235, 2016.

[49] M. Burger, "A framework for the construction of level set methods for shape optimization and reconstruction," *Interfaces and Free boundaries*, vol. 5, no. 3, pp. 301–329, 2003.

[50] C. Y. Kao, S. Osher, and E. Yablonovitch, "Maximizing band gaps in two-dimensional photonic crystals by using level set methods," *Applied Physics B*, vol. 81, no. 2-3, pp. 235–244, 2005.

[51] J. Mockus, *Bayesian approach to global optimization: theory and applications*. Springer Science & Business Media, 2012, vol. 37.

[52] S. D. Campbell, D. Sell, R. P. Jenkins, E. B. Whiting, J. A. Fan, and D. H. Werner, "Review of numerical optimization techniques for meta-device design," *Optical Materials Express*, vol. 9, no. 4, pp. 1842–1863, 2019.

[53] C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch, "Adjoint shape optimization applied to electromagnetic design," *Optics Express*, vol. 21, no. 18, pp. 21 693–21 701, 2013.

[54] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, "Inverse design in nanophotonics," *Nature Photonics*, vol. 12, no. 11, p. 659, 2018.

[55] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007, vol. 5.

[56] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[57] J. C. Mak, C. Sideris, J. Jeong, A. Hajimiri, and J. K. Poon, "Binary particle swarm optimized 2× 2 power splitters in a standard foundry silicon photonic platform," *Optics Letters*, vol. 41, no. 16, pp. 3868–3871, 2016.

[58] D. Dai and S. He, "Optimization of ultracompact polarization-insensitive multimode interference couplers based on si nanowire waveguides," *IEEE Photonics Technology Letters*, vol. 18, no. 19, pp. 2017–2019, 2006.

[59] V. Ganapati, O. D. Miller, and E. Yablonovitch, "Light trapping textures designed by electromagnetic optimization for subwavelength thick solar cells," *IEEE Journal of Photovoltaics*, vol. 4, no. 1, pp. 175–182, 2013.

[60] V. E. Ferry, M. A. Verschuuren, M. C. v. Lare, R. E. Schropp, H. A. Atwater, and A. Polman, "Optimized spatial correlations for broadband light trapping nanopatterns in high efficiency ultrathin film a-si: H solar cells," *Nano Letters*, vol. 11, no. 10, pp. 4239–4245, 2011.

[61] P. R. Wiecha, A. Arbouet, C. Girard, A. Lecestre, G. Larrieu, and V. Paillard, "Evolutionary multi-objective optimization of colour pixels based on dielectric nanoantennas," *Nature Nanotechnology*, vol. 12, no. 2, p. 163, 2017.

[62] Z. Jin, S. Mei, S. Chen, Y. Li, C. Zhang, Y. He, X. Yu, C. Yu, J. K. Yang, and B. Luk'yanchuk, "Complex inverse design of meta-optics by segmented hierarchical evolutionary algorithm," *ACS Nano*, vol. 13, no. 1, pp. 821–829, 2019.

[63] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[64] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[65] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[66] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

[67] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995, ISBN: 0198538642.

[68] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of Electronic Imaging*, vol. 16, no. 4, p. 049 901, 2007.

[69] A. K. Joshi, "Natural language processing," *Science*, vol. 253, no. 5025, pp. 1242–1249, 1991.

[70] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 129–136.

[71] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, pp. 248–255, ISBN: 1424439922.

[72] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732.

[73] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th annual international conference on machine learning*, ACM, pp. 873–880, ISBN: 1605585165.

[74] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105.

[75] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[76] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[77] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, and M. Lanctot, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[78] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, and A. Bolton, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

[80] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

[81] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

[82] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on control systems technology*, vol. 15, no. 3, p. 566, 2007.

[83] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[84] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[85] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[86] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[87] B. Sanchez-Lengeling and A. Aspuru-Guzik, "Inverse molecular design using machine learning: Generative models for matter engineering," *Science*, vol. 361, no. 6400, pp. 360–365, 2018.

[88] A. Lusci, G. Pollastri, and P. Baldi, "Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules," *Journal of Chemical Information and Modeling*, vol. 53, no. 7, pp. 1563–1575, 2013.

[89] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, "Applications of deep learning in biomedicine," *Molecular Pharmaceutics*, vol. 13, no. 5, pp. 1445–1454, 2016.

[90] G. Hautier, C. C. Fischer, A. Jain, T. Mueller, and G. Ceder, "Finding nature's missing ternary oxide compounds using machine learning and density functional theory," *Chemistry of Materials*, vol. 22, no. 12, pp. 3762–3767, 2010.

[91] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning," *Physica D: Nonlinear Phenomena*, vol. 22, no. 1-3, pp. 187–204, 1986.

[92] N. Wang, R. Liu, N. Asmare, C.-H. Chu, and A. F. Sarioglu, "Processing code-multiplexed coulter signals via deep convolutional neural networks," *Lab on a Chip*, vol. 19, no. 19, pp. 3292–3304, 2019.

[93] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annual Review of Biomedical Engineering*, vol. 19, pp. 221–248, 2017.

[94] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of dna-and rna-binding proteins by deep learning," *Nature Biotechnology*, vol. 33, no. 8, p. 831, 2015.

[95] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, "Deep learning for computational biology," *Molecular Systems Biology*, vol. 12, no. 7, p. 878, 2016.

[96] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[97] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, no. 6325, pp. 602–606, 2017.

[98] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics*, vol. 13, no. 5, p. 431, 2017.

[99] L. Wang, "Discovering phase transitions with unsupervised learning," *Physical Review B*, vol. 94, no. 19, p. 195 105, 2016.

[100] Z. Liu, S. P. Rodrigues, and W. Cai, "Simulating the ising model with a deep convolutional generative adversarial network," *arXiv preprint arXiv:1710.04987*, 2017.

[101] N. Wiebe, A. Kapoor, and K. M. Svore, "Quantum deep learning," *arXiv preprint arXiv:1412.3489*, 2014.

[102] T. Zahavy, A. Dikopoltsev, D. Moss, G. I. Haham, O. Cohen, S. Mannor, and M. Segev, "Deep learning reconstruction of ultrashort pulses," *Optica*, vol. 5, no. 5, pp. 666–673, 2018.

[103] H. Kabir, Y. Wang, M. Yu, and Q.-J. Zhang, "Neural network inverse modeling and applications to microwave filter design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 4, pp. 867–879, 2008.

[104] T. Zhang, J. Wang, Q. Liu, J. Zhou, J. Dai, X. Han, Y. Zhou, and K. Xu, "Efficient spectrum prediction and inverse design for plasmonic waveguide systems based on artificial neural networks," *Photonics Research*, vol. 7, no. 3, pp. 368–380, 2019.

[105] G. Alagappan and C. E. Png, "Modal classification in optical waveguides using deep learning," *Journal of Modern Optics*, vol. 66, no. 5, pp. 557–561, 2019.

[106] Y. Rivenson, Z. Göröcs, H. Günaydin, Y. Zhang, H. Wang, and A. Ozcan, "Deep learning microscopy," *Optica*, vol. 4, no. 11, pp. 1437–1443, 2017.

[107] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, vol. 11, no. 7, pp. 441–446, 2017.

[108] J. Feldmann, N. Youngblood, C. Wright, H. Bhaskaran, and W. Pernice, "All-optical spiking neurosynaptic networks with self-learning capabilities," *Nature*, vol. 569, no. 7755, p. 208, 2019.

[109] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," *Science*, vol. 361, no. 6406, pp. 1004–1008, 2018.

[110] C. Qian, X. Lin, X. Lin, J. Xu, Y. Sun, E. Li, B. Zhang, and H. Chen, "Performing optical logic operations by a diffractive neural network," *Light: Science & Applications*, vol. 9, no. 1, p. 59, 2020.

[111] Y. Luo, D. Mengu, N. T. Yardimci, Y. Rivenson, M. Veli, M. Jarrahi, and A. Ozcan, "Design of task-specific optical systems using broadband diffractive neural networks," *Light: Science & Applications*, vol. 8, no. 1, p. 112, 2019.

[112] S. Lan, S. P. Rodrigues, M. Taghinejad, and W. Cai, "Dark plasmonic modes in diatomic gratings for plasmoelectronics," *Laser & Photonics Reviews*, vol. 11, no. 2, p. 1 600 312, 2017.

[113] H. Chen, Y. Weng, D. Xu, N. V. Tabiryan, and S.-T. Wu, "Beam steering for virtual/augmented reality displays with a cycloidal diffractive waveplate," *Optics Express*, vol. 24, no. 7, pp. 7287–7298, 2016.

[114] Y.-H. Lee, G. Tan, T. Zhan, Y. Weng, G. Liu, F. Gou, F. Peng, N. V. Tabiryan, S. Gauza, and S.-T. Wu, "Recent progress in pancharatnam–berry phase optical elements and the applications for virtual/augmented realities," *Optical Data Processing and Storage*, vol. 3, no. 1, pp. 79–88, 2017.

[115] S. An, C. Fowler, B. Zheng, M. Y. Shalaginov, H. Tang, H. Li, L. Zhou, J. Ding, A. M. Agarwal, and C. Rivero-Baleine, "A deep learning approach for objective-driven all-dielectric metasurface design," *ACS Photonics*, vol. 6, no. 12, pp. 3196–3207, 2019.

[116] Y. Zhou, R. Chen, W. Chen, R.-P. Chen, and Y. Ma, "Optical analog computing devices designed by deep neural network," *Optics Communications*, vol. 458, p. 124 674, 2020.

[117] S. So, J. Mun, and J. Rho, "Simultaneous inverse design of materials and structures via deep learning: Demonstration of dipole resonance engineering using core–shell nanoparticles," *ACS Applied Materials & Interfaces*, vol. 11, no. 27, pp. 24 264–24 268, 2019.

[118] S. Inampudi and H. Mosallaei, "Neural network based design of metagratings," *Applied Physics Letters*, vol. 112, no. 24, p. 241 102, 2018.

[119] Y. Long, J. Ren, Y. Li, and H. Chen, "Inverse design of photonic topological state via machine learning," *Applied Physics Letters*, vol. 114, no. 18, p. 181 105, 2019.

[120] L. Gao, X. Li, D. Liu, L. Wang, and Z. Yu, "A bidirectional deep neural network for accurate silicon color design," *Advanced Materials*, vol. 31, no. 51, p. 1 905 467, 2019.

[121] M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, and K. Parsons, "Deep neural network inverse design of integrated photonic power splitters," *Scientific Reports*, vol. 9, no. 1, p. 1368, 2019.

[122] T. Asano and S. Noda, "Optimization of photonic crystal nanocavities based on deep learning," *Optics Express*, vol. 26, no. 25, pp. 32 704–32 717, 2018.

[123] S. Abadal, T.-J. Cui, T. Low, and J. Georgiou, "Programmable metamaterials for software-defined electromagnetic control: Circuits, systems, and architectures," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 1, pp. 6–19, 2020.

[124] J. A. Hodge, K. V. Mishra, and A. I. Zaghloul, "Multidiscriminator distributed generative model for multi-layer rf metasurface discovery," in *IEEE Global Conference on Signal and Information Processing*.

[125] Y. Tang, K. Kojima, T. Koike-Akino, Y. Wang, P. Wu, M. Tahersima, D. Jha, K. Parsons, and M. Qi, "Generative deep learning model for a multi-level nano-optic broadband power splitter," in *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, IEEE, pp. 1–3, ISBN: 1943580715.

[126] K. Kojima, M. H. Tahersima, T. Koike-Akino, D. Jha, Y. Tang, Y. Wang, K. Parsons, F. Sang, and J. Klamkin, "Deep neural networks for designing integrated photonics," in *Optical Fiber Communication Conference*, Optical Society of America, Th1A. 6.

[127] A.-P. Blanchard-Dionne and O. J. Martin, "Successive training of a generative adversarial network for the design of an optical cloak," *OSA Continuum*, vol. 4, no. 1, pp. 87–95, 2021.

[128] J. Jiang, D. Sell, S. Hoyer, J. Hickey, J. Yang, and J. A. Fan, "Free-form diffractive metagrating design based on generative adversarial networks," *ACS Nano*, vol. 13, no. 8, pp. 8872–8878, 2019.

[129] Z. A. Kudyshev, A. V. Kildishev, V. M. Shalaev, and A. Boltasseva, "Machine-learning-assisted metasurface design for high-efficiency thermal emitter optimization," *Applied Physics Reviews*, vol. 7, no. 2, p. 021 407, 2020.

[130] J. Jiang and J. A. Fan, "Global optimization of dielectric metasurfaces using a physics-driven neural network," *Nano Letters*, vol. 19, pp. 5366–5372, 2019.

[131] R. S. Hegde, "Photonics inverse design: Pairing deep neural networks with evolutionary algorithms," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–8, 2019.

[132] S. S. Panda and R. S. Hegde, "Transmission-mode all-dielectric metasurface color filter arrays designed by evolutionary search," *Journal of Nanophotonics*, vol. 14, no. 1, p. 016 014, 2020.

[133] W. Ma, Z. Liu, Z. A. Kudyshev, A. Boltasseva, W. Cai, and Y. Liu, "Deep learning for the design of photonic structures," *Nature Photonics*, vol. 15, pp. 77–90, 2021.

[134] Z. Liu, D. Zhu, L. Raju, and W. Cai, "Tackling photonic inverse design with machine learning," *Advanced Science*, vol. 8, no. 5, p. 2 002 923, 2021.

[135] Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, and W. Cai, "Generative model for the inverse design of metasurfaces," *Nano Letters*, vol. 18, no. 10, pp. 6570–6576, 2018.

[136] Z. Liu, L. Raju, D. Zhu, and W. Cai, "A hybrid strategy for the discovery and design of photonic structures," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 1, pp. 126–135, 2020.

[137] Z. Liu, D. Zhu, K.-T. Lee, A. S. Kim, L. Raju, and W. Cai, "Compounding meta-atoms into metamolecules with hybrid artificial intelligence techniques," *Advanced Materials*, vol. 32, no. 6, p. 1 904 790, 2020.

[138] D. Zhu, Z. Liu, L. Raju, A. S. Kim, and W. Cai, "Building multifunctional metasystems via algorithmic construction," *ACS Nano*, vol. 15, no. 2, pp. 2318–2326, 2021.

[139] Z. Liu, Z. Zhu, and W. Cai, "Topological encoding method for data-driven photonics inverse design," *Optics Express*, vol. 28, no. 4, pp. 4825–4835, 2020.

[140] D. Gostimirovic and N. Y. Winnie, "An open-source artificial neural network model for polarization-insensitive silicon-on-insulator subwavelength grating couplers," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 25, no. 3, pp. 1–5, 2018.

[141] P. R. Wiecha and O. L. Muskens, "Deep learning meets nanophotonics: A generalized accurate predictor for near fields and far fields of arbitrary 3d nanostructures," *Nano Letters*, vol. 20, no. 1, pp. 329–338, 2020.

[142] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[143] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[144] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680.

[145] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, and P. Simard, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the Statistical Mechanics Perspective*, vol. 261, p. 276, 1995.

[146] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, pp. 214–223.

[147] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, pp. 5769–5779.

[148] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic Programming and Evolvable Machines*, vol. 8, no. 2, pp. 131–162, 2007.

[149] M. A. Potter and K. A. D. Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, no. 1, pp. 1–29, 2000.

[150] A. Migdalas, P. M. Pardalos, and P. Värbrand, *Multilevel optimization: algorithms and applications*. Springer Science & Business Media, 2013, vol. 20.

[151] F. Wyrowski and O. Bryngdahl, "Iterative fourier-transform algorithm applied to computer holography," *Journal of the Optical Society of America A*, vol. 5, no. 7, pp. 1058–1065, 1988.

[152] F. Roddier and C. Roddier, "Wavefront reconstruction using iterative fourier transforms," *Applied Optics*, vol. 30, no. 11, pp. 1325–1327, 1991.

[153] M. Moharam, E. B. Grann, D. A. Pommet, and T. Gaylord, "Formulation for stable and efficient implementation of the rigorous coupled-wave analysis of binary gratings," *Journal of the Optical Society of America A*, vol. 12, no. 5, pp. 1068–1076, 1995.

[154] R. Zhang, Z. Liu, K. Yang, T. Liu, W. Cai, and L. Milor, "A library based on deep neural networks for modeling the degradation of FinFET SRAM performance metrics due to aging," *Microelectronics Reliability*, vol. 100, p. 113 486, 2019.

[155] R. Zhang, Z. Liu, K. Yang, T. Liu, W. Cai, and L. Milor, "Inverse design of FinFET SRAM cells," in *2020 IEEE International Reliability Physics Symposium (IRPS)*, IEEE, 2020, pp. 1–6.

[156] R. Zhang, Z. Liu, K. Yang, T. Liu, W. Cai, and L. Milor, "Impact of front-end wearout mechanisms on finfet sram soft error rate," *Microelectronics Reliability*, vol. 100, p. 113 487, 2019.

[157] R. Zhang, K. Yang, Z. Liu, T. Liu, W. Cai, and L. Milor, "A comprehensive framework for analysis of time-dependent performance-reliability degradation of sram cache memory," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2021.

[158] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances*, vol. 3, no. 4, e1602614, 2017.

[159] J. Han, A. Jentzen, and E. Weinan, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, 2018.

[160] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[161] J. Jacobsen, A. Smeulders, and E. Oyallon, "I-revnet: Deep invertible networks," in *ICLR 2018-International Conference on Learning Representations*, 2018.

VITA

Zhaocheng Liu received his B.S. degree from Nankai University in Physics in 2015, and his M.S. degree from Georgia Institute of Technology in electrical engineering in 2016, respectively. He is currently pursuing his Ph.D. degree at Georgia Institute of Techonology. His research interests include photonics, physical modelling, machine learning, and optimization methods.